Jurassic News



Silicon Graphics INDY

Super QL

Lo sviluppo del linguaggio C

Il mio Z80

Nosferatus

L'indecibilità

Geriatric Linux

Retrocomputer Magazine

Anno 11 - Numero 57 - Febbraio 2016

Jurassic News
Rivista aperiodica di Retrocomputer

Coordinatore editoriale:
Tullio Nicolussi [Tn]

Redazione:
redazione@jurassicnews.com

Hanno collaborato a questo numero:
Lorenzo [L2]
Salvatore Macomer [Sm]
Sonicher [Sn]
Besdelsec [Bs]
Lorenzo Paolini [Lp]
Mario Raspanti
Gizmo
Emery Fletcher
Damiano Cavicchio

Diffusione:
Lettura on-line sul sito o attraverso il servizio Issuu.com; il download è disponibile per gli utenti registrati.

Sito Web:
www.jurassicnews.com

Contatti:
info@jurassicnews.com

Jurassic News
promuove la libera circolazione delle idee

Jurassic News

E' una fanzine dedicata al retro-computing nella più ampia accezione del termine. Gli articoli trattano in generale dell'informatica a partire dai primi anni '80 e si spingono fino ...all'altro ieri.

La pubblicazione ha carattere puramente amatoriale e didattico, tutte le informazioni sono tratte da materiale originale dell'epoca o raccolte su Internet.

La redazione e gli autori degli articoli non si assumono nessuna responsabilità in merito alla correttezza delle informazioni riportate o nei confronti di eventuali danni derivanti dall'applicazione di quanto appreso sulla rivista.

Il contenuto degli articoli è frutto delle conoscenze, esperienze personali e opinioni dei singoli autori; possono pertanto essere talvolta non precise o differire da fonti "ufficiose" come Wikipedia e siti Web specializzati.

Sono gradite segnalazioni di errori, imprecisioni o errate informazioni che possono, a discrezione della redazione, essere oggetto di errata-corrige in fascicoli successivi.

Scrivere a:
redazione@jurassicnews.com
dettagliando il più possibile l'argomentazione.

Veterani cercasi.

Come è lontana l'Inghilterra!

Il museo nazionale "The National Museum of Computing (TNMOC)" con sede a Londra, nel famoso Bletchley Park, ha fra le sue iniziative dei veri e propri percorsi di retro-informatica che sono seguiti annualmente da circa 4500 studenti delle superiori.
Numerosissimi i progetti, le iniziative e importanti anche gli sponsor (l'ultimo è Fujitsu) che consentono di portare avanti iniziative di un certo spessore.
Fin qui quello che manca in Italia, anche se le iniziative private ci sono e sono degne del massimo rispetto.



La collezione ricca di reperti (uno per tutti il famosissimo Colossus) ha come massima numerosità il famoso micro BBC, uscito nel 1981 e diffusissimo come macchina da studio sia nelle scuole che nelle case inglesi.
Ora questa numerosissima collezione di micro BBC abbisogna di cure e i volontari non sono mai abbastanza.
Ecco dunque che il museo cerca veterani (proprio così li chiama) che abbiano dimestichezza con la macchina nei suoi aspetti, hardware, software e periferiche varie.
L'obiettivo è la cultura.

E le istituzioni pubbliche italiane?
Assenti.
C.v.d.

Sommario

Jurassic News - Il contenuto di questo fascicolo

OK, il prezzo è giusto!



di Tullio Nicolussi

Periadicamente se ne parlo: quol è il valore di un retra computer e per canseguenza qual è il prezzo equa da sbarsare per acquistarne una?

Nanostante ci sia qualche tentativa di campilare un listina identificanda per agni oggetta una fascia di prezza, tali iniziative sona destinate a suscitare l'ilarità di malti e l'irritazione di altrettanti canoscitari e patenziali appassianati di retra informatica.

Il prablema nan è l'impreparaziane di chi campila i listini (svariani a parte). Il fatta è che è oncora un mercato malta legato alla nazianalità. Cioè un sistema camune in Italia, od esempio l'M24, patrebbe essere raro negli States e quindi spuntare un buan gap di differenziale di prezzo fra le due nozioni.

Spesso anch'ia mi sona trovota davanti ol dilemma se comprore a desistere, al punto che come regala generale mi sona ripromesso di non acquistore mai se il prezza richiesto supera i 50 Euro, qualunque sia la stato dell'opporecchio propasto.

50 Eura è una saglio psicolagica più che un dato di valenza rispetta ol prodotta. Tuttavia sarei disposto a più di una ecceziane se la richiesta riguardasse un pezzo particalarmente significativa. Nan dica un pezzo "rara", per il quale le quatazioni esulano dall'aspetta retra-camputer e sfaciana nel campa della stima dei reperti artistici. Un Apple I nan si discute, avviamente!

Già per un Lisa patrei anche arrivare ai 100 Eura, ben canscia di star prendenda un qualcasa che non funziona a che è malto difficile da riparore un giorna che avesse deciso di "passare o miglior vita". Nan spenderei moi 100 Euro per un C64, nemmena mancosse alla mia callezione, a per una Spectrum...

Va bene, queste due ultime menzianote sono macchine comuni, nel senso che si può ragionevalmente rifiutare l'acquisto oggi per travare domani lo stessa prodotto a prezzo equivalente o inferiore. Quindi una qualche idea della fascia di prezzo di un certo reperto la abbioma: farse ce la siamo fotta mediando le aste su eBay? Può essere:

pensa che chiunque ci faccia un gira su eBoy primo di vendere/comprare qualsiasi casa.

Malti cammentano il valare presunto della lara collezione, appeno qualche incauta venditore posta un orticolo a prezzo esagerato. Queste "uscite dol seminota", che apparentemente sembrano tutte frutto dei tentativi di utenti nan esperti del comparto, in realtà qualche valta mi fanno sospettare diversamente.

Ha una apiniane in propasita e cioè che ci siana due specie di venditori "fuori bersaglia": gli incansapevali e i furbastri.

Dei primi c'è paco da dire: ci provana a vendere il lara bravo C64 a 200 Eura e il lara bravissima Amiga 500 a pesa d'oro. Ci provana e di solito sona sbeffeggiati con lazzi e sberleffi degni della migliore gagno mediatica.

I secondi, ciaè i furbastri, mi preoccupana un pochino di più. Ci sono quelli (almena un paio in Italia) che vendano le macchine o pezzi. A seguire un attimo le lara aste si capisce che hanna presa un computer, paniamo un Amiga, l'hanno smontato, talti i chip e tentano di realizzare un totale lorgamente superiore a quello che ragionevolmente (e lo sanno bene) potrebbero ricavore do uno macchina intonsa.

E' focile fare due conti: so che potrei vendere diciama a 100 Euro quell'Amiga completo con una certa dotozione di software e con i suoi manuoli. Ma invece ne ricavo 20 pezzi (per non esogerare), che venduti a 15 euro cadauno (casa vuoi che siana 15 Eura quanda ti serve praprio quel particalare?), ne ricava 300 di Eura e camunque 150 se ne vendo anche sola 10.

Ottenere 20 pezzi da un Amiga non è difficile se ci si mette di impegna, basta camínciare can l'invalucra, due manuali, un joystick, l'alimentatare, un disca esterna, tre giachi in confeziane originale, un set campleta di 50 floppy usati can giochi a go-go, qualche rivista... E poi si comincia a smantore e vio le ROM di kickstort (quante ne obbioma viste su eBay?), la RAM, i chip Paula, etc...,

Spero veramente di non stare dando strane idee a qualcuna...

Ammetto che recuperare un set di chip Amigo potrebbe risolvere il prablemo del sistema che abbiama in casa e che risulta mancante praprio di ciò che viene afferta, ma dal punta di vista del canservatore non varrei mai vedere un sistema funzianante smembrata per bramasia di qualche Eura!

Inaltre c'è un effetta callaterale spiacevale: quanda tizia ha venduta i pezzi più appetibili (e più cari) è facile che non si tenga in casa un case o una tastiera semi cannibalizzata, con il risultota che la rottamazione taglie definitivamente dalla circolaziane quella che poteva essere un degno parteciponte ad una colleziane.

Credo in ultima analisi che la lievitaziane dei prezzi di qualche reperto, dichiaratamente nan funzionante, sia davuta anche o questa pratica.

Che fare?

Nan ho soluzioni. Patremmo tutti non comprare pezzi singoli obbligondo chi vende o quotare "vuota per pieno" l'intero macchina. Patremma stabilire ad esempia una quotazione massimo per il sistemo nan funzianante, qualunque esso sia (nei limiti dello possibilitò di effettuare una clossificaziane amagenea).

Tutte cose facili a dirsi ma di difficile realizzaziane.

La casa migliore però è il campartamento etico di ciascuna di nai e che creda debba far parte del bagaglio dell'appassianato di retro camputer, casì came di qualsiasi altra genere di callezianisma.

(=)

Silicon Graphics - INDY



di Mario Raspanti

Incipit

E' ormai passato molto tempo dall'articolo sulla Silicon Graphics Indigo, apparso su Jurassic News No.27. Oggi vediamo la macchina che a questo modello ha fatto seguito, e che insieme alla successiva O2 ha forse rappresentato il modello più popolare di casa Silicon Graphics (o SGI).

Il contesto

Siamo nel 1993. Presidente del Consiglio è Carlo Azeglio Ciampi. Alla Casa Bianca si insedia Bill Clinton. Al cinema esce Jurassic Park, cui la stessa SGI ha pesantemente contribuito e che apparirà nella sua pubblicità dell'epoca (Building a better dinosaur). La Intel ha appena presentato il nuovo processore Pentium 60 MHz, e i PC di fascia alta già in circolazione sono dei 486DX-33 o 486DX2-66, di solito con 1 o 2 MBytes di RAM. Il sistema operativo corrente è Windows 3.1, ma i giochi non sono ancora fatti e nel dubbio la stessa Microsoft gli affianca anche la prima versione di Windows NT, anch'essa chiamata 3.1, e la versione 2.1 di OS/2, lenta e pesante (OS/2 Warp è ancora lontano). Nel campo professionale ci sono le workstation DEC 3000, la IBM RS/6000 e la SUN SparcStation 10.

In questo contesto la Silicon Graphics sostituisce la "piccola" di casa, ossia la sua prima e finora unica macchina desktop, di-

videndola in due madelli:

- *La Indigo2, che della Indigo eredita il name e la destinaziane professianale, è un enarme desktop/deskside (puà essere montata anche in verticale can due appositi sostegni, farniti di serie) da circa 50 x 50 x 15 cm e 20 Kg di pesa, di un singalare calare verde/blu. La macchina è rabustissima, performante e malta espandibile; in seguita conascerà una secanda giavinezza con una secanda serie madificota, la Indigo2 Impact, can nuavi processori e nuove schede grafiche, e can la corrozzeria calore fucsia (!).*
- *La Indy, più snello, leggera ed economica, è meno espandibile ma più versatile e giò dotata di serie di una impressiononte quontità di interfacce e cannessioni.*

La macchina offre di serie un processore RISC ad almeno 100 MHz, grafica od olta risoluziane con 16 milioni di colari ed occeleraziane 3D, quottro canoli audio, digitalizzatari audio e video incorporati, due diversi stondord Ethernet, ISDN, due porte seriali, una SCSI, una parallelo ed un sistema aperotiva Unix-based can uno interfaccio grofica evaluta.

Mentre è facile o posteriori comprendere la destinaziane cammerciale della Indiga2, la Indy occupa una pasizione più sfumata. A chi era indirizzata questa macchina, più brillante della sarella ma tutto sommata meno performante e ancar mena espandibile, ad un prezza che (seppur bassa in termini SGI) la separava canunque nettamente dai PC di fascia alta? Prabahilmente la Indy è stata in qualche mada una macchina-esca, pensata per attirare verso i pradatti SGI una clientela più vosta ed eterogenea di quello già consolidata. Senza dubbio, il gronde successo che ebbe (all'estero) nei computer center delle università ed affini ovvicinò al marchia Silican Graphics un vasta pubblica di studenti e futuri professionisti e, come vedremo, creà un vasta movimento di interesse intorna alla macchino.

La macchina

La Indy si presenta quasi come un tipica desktap dell'epaca, ossia un parallelepipeda piatta a sviluppa orizzantale, di circa 40x35 cm ed alto circa 8 cm, cancepita per stare satta il manitar (Figura 2). Non propria tipica, perché la macchina ha diverse peculiarità di design: innanzitutta il calare è un blu acceso, mavimentato da un particolare effetto granito, mentre tastiera ed accessori hanna la stessa finitura ma in calare grigio. Un gradina attraversa obliquamente lo macchina, mavimentandone la farma. Di una estrema pulizia il design del frontale, dove nan troviamo nessuna operturo ma solo il tasto di accensione con la relativo spia (Figura_3), un minuscalo tasto di reset nascosta in una sconalatura e due tosti triongalari per la regolazione del volume (!) dell'altaparlante interna. Non è previsto un alloggiamento interno per un CD o una unità o nostro magnetico, che vanno collegoti esternamente; è invece possibile montare un floppy drive interno, cui si occede dal fianca destro della mocchina (Vedi la fato di apertura) e che deve overe l'espulsione sof-





Come al solito in caso Silicon la stessa macchina era disponibile in vorie soluzioni e configurozioni. Per la CPU erono disponibili non meno di dodici CPU diverse, tutti processori MIPS RISC con velocità da 100 o 200 MHz, qui elencati in ordine crescente di potenza:

R4000PC-100
R4000SC-100
R4600PC-100 e 133
R4600SC-133
R4400SC-100, 150, 175 e 200
R5000PC-150
R5000SC-150 e 180

La R4600 venne introdotta dopo la R4400, ma resta meno performante di quest'ultima. Le versioni PC sono prive di una memoria cache interna e, a parità di clock, sono sensibilmente più lente delle versioni SC. La RAM (Figura_5) era costituita da comuni moduli SIMM tipo PS/2 con parità, 36 bit su 72 pin, usoti anche sulla Indigo R4000 e sulla Indigo2 e con una capacità fino a 32 MBytes per modulo. Per la porticolare architettura i moduli devono essere inseriti in gruppi di quottro per volta.

Le opzioni grafiche erano invece tre e, con le loro prestazioni tutto sommato vicine a quelle della precedente Indigo, sembrano stra-

tware (à la macintosh), perché non è previsto il solito pulsante. Come si vede, non è l'originalità quella che manca.

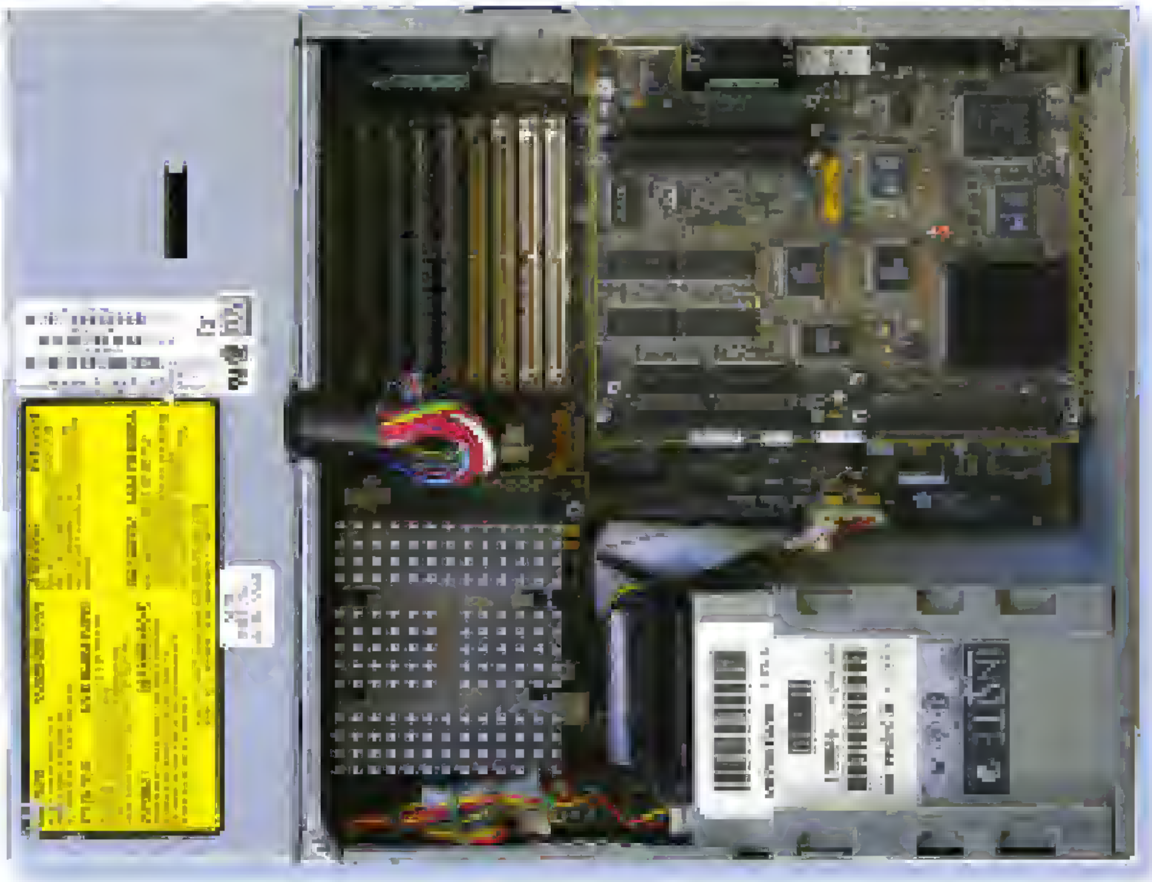
La Indy si apre sfilando verso l'avanti l'intera copertura plastico (detta skin in casa SGI).

Il lato sinistro (vedi Figura_4) è interamente occupato dall'alimentatore, montoto a sbolzo e trattenuto da una sola vite e da un incostro. Sul frontale troviamo in posizione centrale la CPU, montata su una propria schedo e provvisto di una impressionante (per l'epoco) alettatura di raffreddomento, e a destra i dischi, rivolti come si è detto verso il fianco destro. La porte posteriore è occupoto dalla schedo madre, sulla quale è montata orizzontalmente lo scheda video. Resta spozio per altre schede, da montore superiormente a quest'ultima.

L'interno è molto pulito e le parti principali sono tutte rapidamente accessibili. Avendo i pezzi di ricambio, una Indy guasta può essere fatta ripartire in pochi secondi e con un normale cacciavite.

La costruzione sembra invece decisamente gracile in confronto alle altre macchine della stessa casa, tutte costruite come un carro armato. In porticolare il pesante alimentatore sembra decisamente posticcio, ma devo ammettere che non si è mai stoccato.



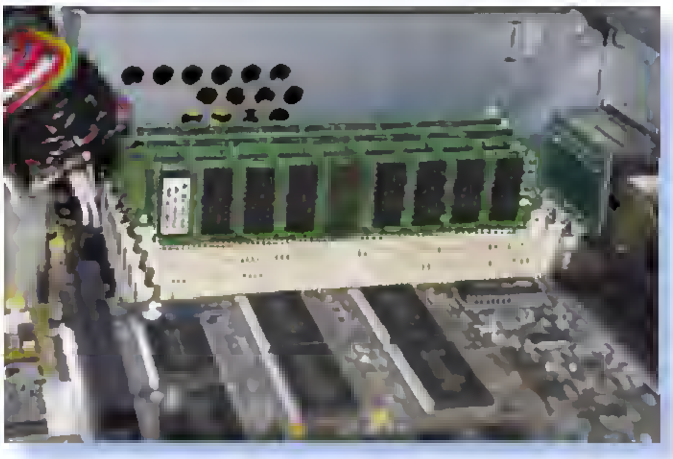


namente sotto tono per una macchina così innovativo:

XL8 8 bpp, 1280x1024 oppure 1024x768, 256 colori
XL24 24 bpp, 1280x1024 oppure 1024x768, 16 milioni di colori
XZ 24 bpp, 1280x1024 oppure 1024x768, 16 milioni di colori e accelerazione 3D.

Nel complesso le prestazioni sono ottime in 2D, ma piuttosto anemiche in 3D. Basti dire che la scheda XZ, l'unica con accelerazione 3D hardware, veniva sconsigliata sulle Indy R5000 perché in questo caso la CPU era più veloce della GPU o gestire le trosformazioni, e lo presenzo di una XZ rallentova la macchina. Naturalmente chi oveva bisogno di prestazioni veramente superiori veniva indirizzato verso un sistema Indigo2.

Tutte le schede gestivano come abbiamo visto due risoluzioni fisse: 1024x768 e 1280x1024. La risoluzione poteva essere cambiota in softwore o automoticamente vio hardware, tramite un contatto nel connettore video che, per i monitor SGI, riconoscevo la periferica video in uso.





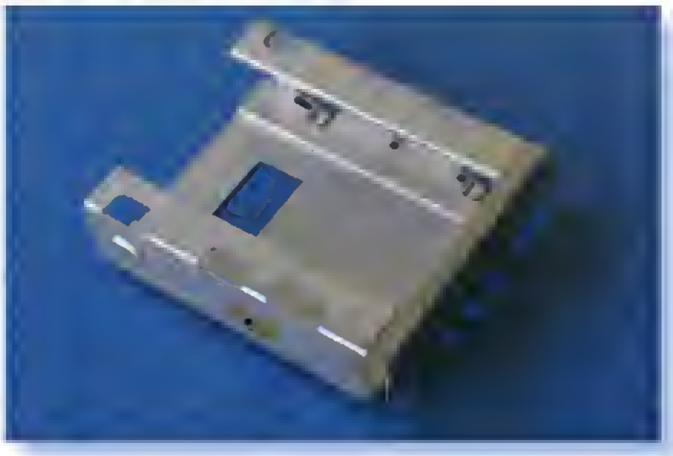
Lo spozio per i dischi (Figura_6) è molto ridotta e la Indy accetta ol massimo due unità a mezza oltezza, agnuno inserite in una proprio slitta a scotto in metolla molta rozionale (Figura_7) e montote sovrapposte. L'unità inferiare di solita cantiene l'hard disk; nella superiore può essere alloggiota un floppy disk o, came nella macchina qui raffigurata, un floptical da 21 MBytes (vedi Jurassic News No. 33). Tutte le unità sano naturalmente SCSI. Quanda si montavana due hard disk veniva cansigliata di limitarsi ad unità da 10.000 RPM piuttasta che da 15.000 RPM, comuni sulle workstatian high-end, per evitare prablemi di surriscaldamento. Nel casa il bay superiore restasse inutilizzato la SGI raccomandova di lasciare camunque inserito lo relativa slitta metallica, che cantribuiva a sappartore il pesa del monitor savrastonte.

Le interfacce

Qui lo Indy si rivelo veromente difficile da battere, e il pannello posteriore (Figura_8) è praticomente uno distesa di connettori. Da sinistra, in basso traviama una fila di cinque prese jack stereo do 3.5 mm: nell'ordine troviamo la presa di cuffia, l'entrata microfana, l'entrata di linea, l'uscita di linea e l'in/aut digitale, ricanfigurabile via software in modo da attenere quattra canali. Sapra a questi la grande presa 13W3 per il monitar (RGB, sync-an-green) e la piccala mini-DIN per la sincranizzaziane degli occhiali sterea (stiamo parlanda di vent'anni fa!).

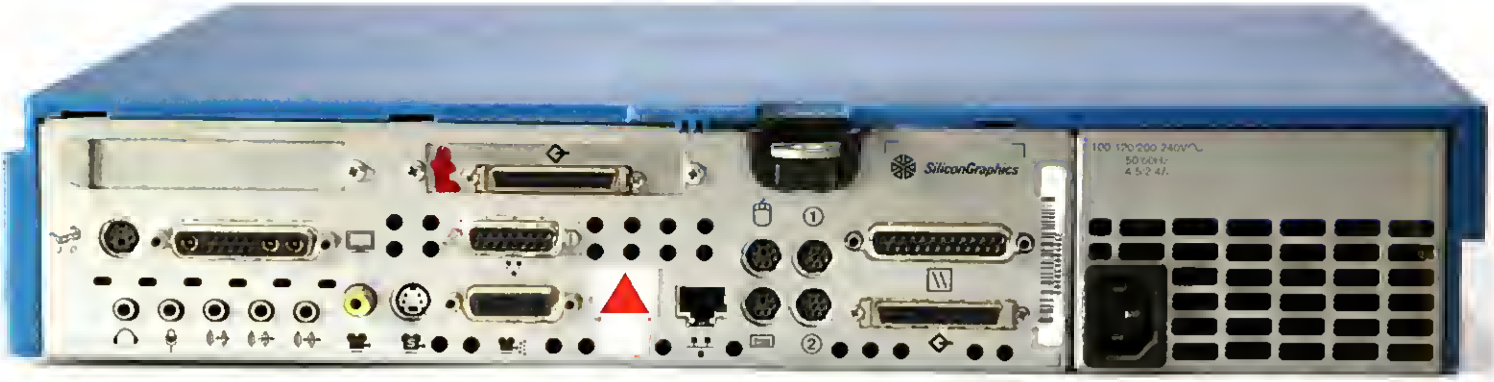
Ricardiama che in casa SGI la presa 13W3 ho un coblaggia leggermente diverso da modello o madella (e anche leggermente diverso da SUN, IBM ecc...) e che un adattotare VGA, necessaria per cannettere un monitar odierno, nan è detta che funzioni al prima calpo.

Procedendo versa destra troviamo in successione tre ingressi videa: un connettore RCA, un ingressa S-Video ed una porta video praprietaria, dedicota od uno speciole IndyCam (più oltre) e vagamente somigliante od un DVI.



Sopro o questa abbiamo la presa D a 15 poli per una rete thick ethernet. Più in alto due tappi coprona le finestre dedicote ad eventuoli schede accessorie.

Procedendo verso destra, una eti-



chetta con un triongolo rosso copre una preso ISDN, che mecconicamente è uguale allo successivo preso di rete RJ45, onde evitare costosi errori nell'inserire i cavi. L'interfaccio di rete è una sola, nel senso che il connettore thick ethernet e il connettore RJ sono in alternativa – o si usa l' uno o l' altro.

Seguono le prese tastiera e mouse, che per la prima volta in casa SGI sono unità standard PS/2, e due porte seriali su connettore mini-DIN, come allora usava anche Apple. Infine un connettore SCSI per unità esterne e, sopra a questo, una porta parallela su connettore D a 25 poli. All'estremità destra infine troviamo l'alimentatore con la sua presa per il cavo di rete.

Serve altro?

Eccome!

Sempre nel campo hardware, la Indy ha ancora un poio di ciliegine da mettere sul gelato.

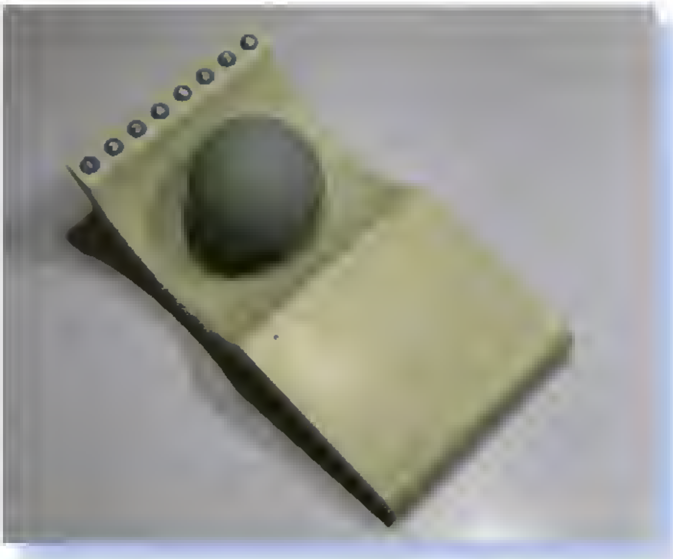
La prima è lo IndyCom (Figura_9), che potrebbe essere considerata la prima WebCom della storia: una piccolo videocamera, color grigio-granito come tutti gli occessori Indy, destinota al teleconferencing (...nel 1993, ricordiamolo) ma anche ompiamente usoto per la produzione di piccoli (e ingombranti) video home-made. Vo ricordoto che la IndyCam può essere collegata soltanto ad una Indy, ma il digitalizzatore audio/video (di serie!) di quest'ultima funziona perfettamente anche con gli altri ingressi, CVBS o S-Video, a cui può essere collegata qualunque sorgente video. Unico problema, produce files non compressi e quindi relotivamente ingombranti, che erano uno costante dannazione degli amministratori nei centri di calcolo. Se proprio si voleva andare sul pesante erano disponibili due diverse schede opzionali, da montare in piggyback sulla scheda video, per la compressione video in tempo reale.

La Indy naturalmente poteva montare anche tutta una gamma di accessori generici, come il mouse tridimensionale Spaceball (Figura_10), array di pulsanti e di manopole, tavolette grafiche ecc.

Infine il sacro Groal dei possessori di Indy, che purtroppo non siamo in grado di mostrarvi, è la scheda Ultra-64. Questa, che ondava montata anch'essa sopra la scheda video, è la parte hardware del sistema di sviluppo per Nintendo 64.

E' naturale che i videogiochi per piccole consolle venissero sviluppati su altri sistemi più potenti e più veloci; la stessa cosa accade anche adesso. Dove è nato SuperMario 64? Adesso lo sapete!

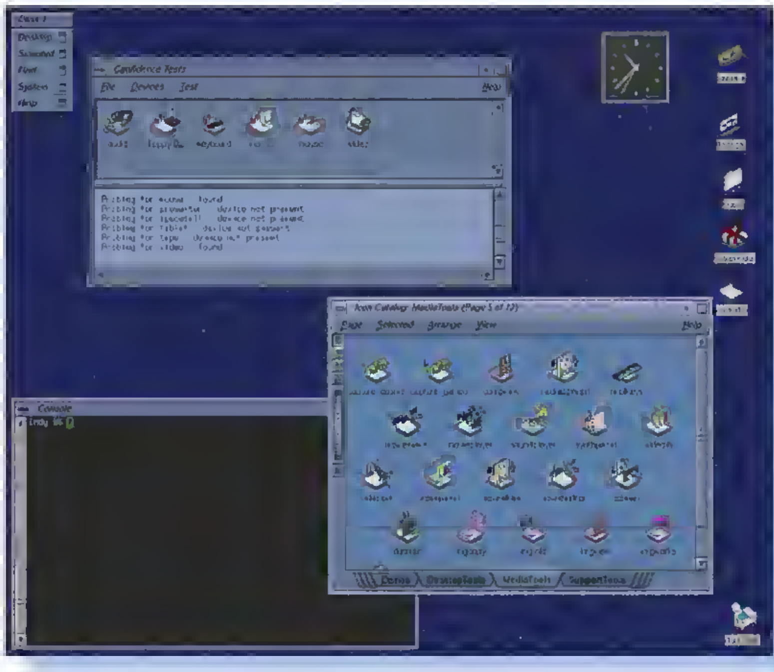


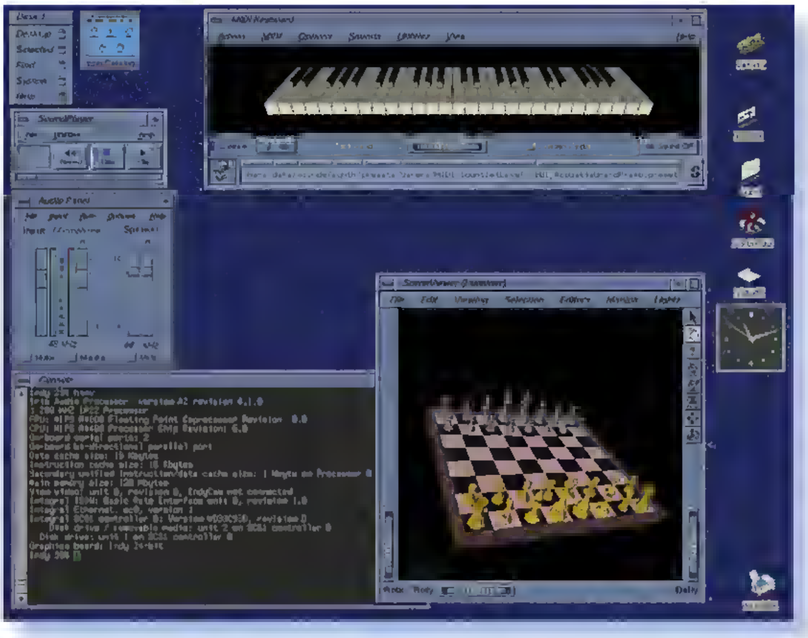


Altri rorissimi accessari (anche se nan fanno porte dell'hardware in sensa stretto) sano la borsa originale per il trosporto della Indy (mentre ricardiamo che non è mai esistito un portatile Silicon Grophics; se ne intravede una fasulla nel film Twister) e una inverasimile valigetta 24 ore o forma di Indy, da cui si distingue salo per la presenza del manica, prodatto -pare- in soli 50 esemplari usando gli stampi ariginali.

Il software

E' avviamente Irix, versione Silicon Graphics di Unix, che come la moggior porte dei sistemi Unix ha prestaziani elevatissime e non si pianta mai. Ne sano esistite numerose versiani, alcune molto diffuse ed alcune rimaste allo stadio di sigla; l'esemplare in aggetto manta una versiane 6.2, ma lo Indy è in grado di mantore fina alla maggiar parte delle versiani 6.5.





All'accensione la Indy emette un bip, seguito da un breve stacchetto musicale (una caratteristica di malte macchine Silicon Graphics, ognuna con un motivetta diversa) e dalla camparsa della schermata di welcame. Quanda la macchina ha campletata la diagnostica la spia di accensiane da rossa diventa verde e la Indy affre, per qualche secanda, la passibilità di interromperla per richiedere una diagnastica più apprafandita (una cosa veramente lunga) a effettuare operazioni di sistema, come un upgrade del sistema operativa. Dopadiché carica Irix e presenta la schermata di lagin.

L'estetica della GUI Motif (Figura_11_12) è sobria e persino austera in confronta alle attuali versiani Windows e Linux, ma in queste macchine ogni dettaglio, dai font ai colori, era attentamente studiato per massimizzare la produttività e per nan stancare l'utente durante lunghe are davanti al monitar, non certo per impressionare gli amici. La Indy in esame è una R4400SC-200MHz can una scheda grafica XL24, ossia una configuraziane media-alta per una Indy, ma nan certa paraganabile ad una macchina attuale. Eppure la responsività della GUI è malta maggiare: al tocca del mause le finestre si aprano subita, i programmi partano immediatamente, il feedback è istantaneo. Ci si dimentica presto della clessidra che ci fa campagnia nel nostra salito PC, e saprattutta non si deve riavviare mai. In sensa letterale: la Silican Graphics raccamandava di lasciare sempre accese le sue macchine, e del resto l'unica mativa per spegnerle era un guasta hardware.



Satto la GUI c'è, come abbiama detto, Unix, can tutto quel che comparta ma anche con alcune estensiani. Una esclusività Irix è un



comodo Toolchest, in alto a sinistra, con i principali comandi di sistema, tra cui una serie di approfonditi Confidence tests per tutte le periferiche presenti e un Icon catalog che raccoglie, ordinatamente per tabs, decine e decine di applicazioni. Possiamo ricordare anche alcuni utili comandi aggiuntivi da terminale, come hinv per l'inventario hardware o chkconfig per avviare/fermare individualmente i daemons. Vale per la Indy tutto quello che nel numero 27 è stato detto sul software della Indigo, a cui vi rimando.

A livello applicativo devo ricordare l'esistenza, sempre di serie, di moltissimo software multimediale, convertitori di files, generatori musicali, di visualizzatori 3D ecc. (Figura_13); ma anche di software di rete e di teleconferencing, quando questa parola non esisteva neppure. Successivamente è stato reso disponibile anche un emulatore Windows 95, di utilità pratica forse limitata ma comunque impressionante (anche perché invece un emulatore Indy sotto Windows 95 non l'ho mai visto).

Sul fronte software bisogna tener conto, da un lato, della disponibilità di software applicativo di altissimo livello (un esempio per tutti è Wavefront, che nel 1995 verrà acquistato direttamente dalla SGI e da cui nel 1998 nascerà Maya); dall'altro della disponibilità, tipica del mondo Unix, di software gratuito. La velocità della macchina, la qualità del sof-

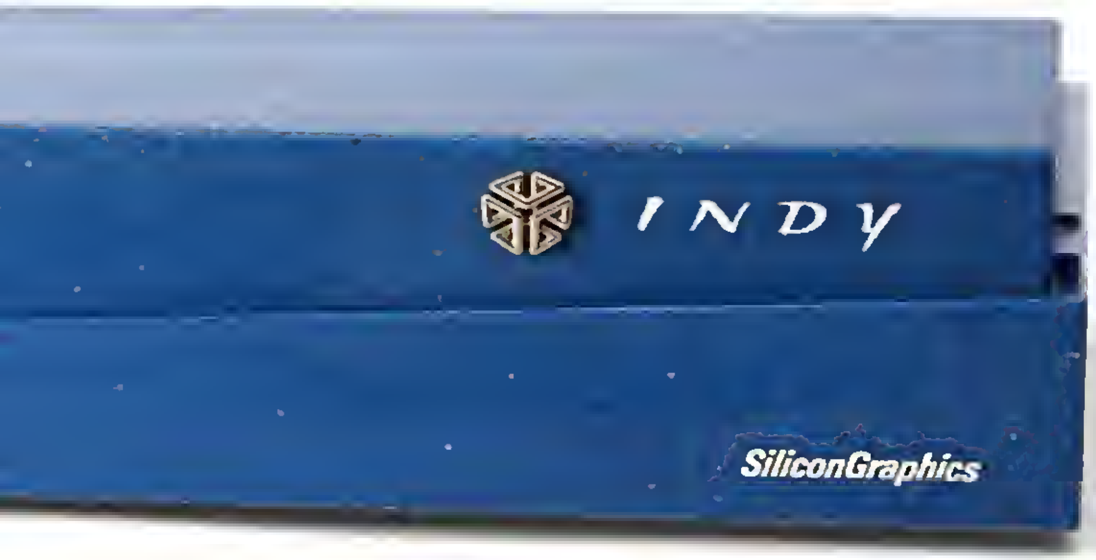


tware, lo facilità di interfacciamenta hanno infatti determinata, alla sua epaca, la sviluppo di una attivissima camunità di utenti Silican Graphics e di un clima culturale malta dinamica e callabarativa, can un attiva scambia di idee e di saftware che, seppur in chiave minare, cantinua ancora oggi. La stessa Silican Graphics incaraggiava questa atteggiamento mediante un contest chiamata IndyZane articolata in cinque settari: sala player games, team games, single user taols, multiple user taols, e best Indy software. Ogni categoria prevedeva una macchino SGI in premia al vincitore e la diffusione del software mediante i CD IndyZane (pare ne siana usciti cinque). Altri CD, contenenti demos di programmi commerciali, utilities e freeware, venivana diffusi dalla Silicon Graphics tra la cammunity degli utenti con il nome di Hat Mix. Tutti questi sona ancara ottivamente ricercoti.

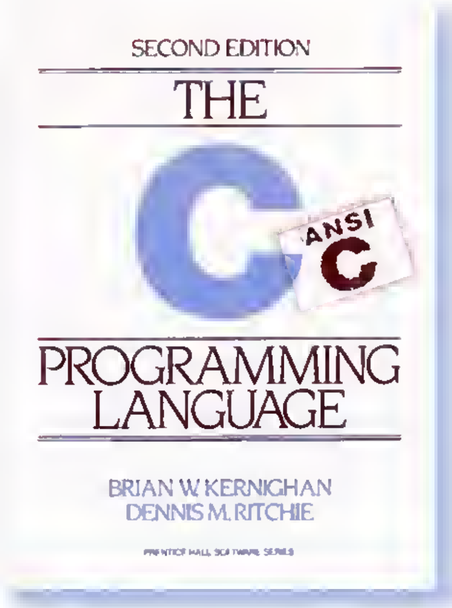
Naturalmente sarebbe inutile cercore Assossin Creed tra il softwore home-mode di vent'onni fa, ma il livella è comunque decisamente brillante. Un esempio è visibile nell'ultima immagine (Figuro_14), che raffigura il vincitore di IndyZone 1994 nella categorio best salo player game: Blix, uno specie di PacMan mappata su una sfera, dave altre a guidare l'avatar davete, man mana che questa si muave, girare anche la sfera in mado da nan perderla di vista.

Sembra semplice, vera? Pravate a giacarci...

(=)



Lo sviluppo del linguaggio C



di Salvatore Macomer

Introduzione.

Il linguaggio C è diffusa came farse nessun altra linguaggia di programmaziane. Gli elementi principali della sua nascita sana nati: progettata pressa i Bell's Laborataries do B. W. Kernighan e Dennis M. Ritchie came evaluziane del linguaggio B, o sua valto derivato dal BCPL, per lo sviluppa del sistema operativo Unix, si impose ben presta sia per le sue caratteristiche che per la sua partobilità.

Questa articola rivisita i passi essenziali della sua nascito e saprottutta dello suo evoluzione ed è basato in particolare su un orticalo scritta da Ritchie in persana nel 2003 per lo Lucent Technology, che nel frattempo ha rilevata i Bell's Labaratories.

Uno sguardo di insieme.

Il C nasce negli anni 1969-1970 in parallela all'evoluzione del sistema aperativo Unix. Il sua ricanascimenta ufficiale si ebbe nel 1977 e nel 1979 quando la partabilità di Unix fu definitivamente dimastrota. Finalmente olla metà degli anni '80 il linguaggio venne standardizzato dal comitato ANSI X3J11, quando ormai esistevano compilotari C per quasi tutte le piattaforme, hame camputer campresi.

Nel 2005 è stato eletto al vertice della classifica dei linguaggi di programmozione più diffusi al mando, scettro che prababilmente detiene oncoro, mentre di fotto tutti o quasi i linguoggi "maderni", si basana sulle sue idee, pur evolvenda l'ambiente di produzione del saftware verso una canceziane più consona olle idee della programmazione od

oggetti.

Il testo "principe" che ne descrive le corotteristiche è il fomoso "libro bionco" scritto doi due outori del linguoggio e che ho come titolo "The C Progromming Longuoge". Viene chiamoto "libro bionco" perché nello prima edizione, coso rimosto invoriato nelle successive, lo copertino è semplicemente bionco con uno gronde C azzurro che spicco nel titolo.

Premesse storiche.

Allo fine degli onni '60 i Bell Telephon Laborotories decisero di uscire dallo join venture che vedevo come oltri ottori il MIT e lo General Electric. Il consorzio aveva come obiettivo lo reolizzozione di un sistemo operotivo multitosking denominoto Multics, progettoto per il moinfrome di Generol Electric GE 645. Lo reolizzozione si stovo però rivelondo irto di ostocoli e lo prospettivo di un riloscio definitivo si ollontonovo sempre di più, oltre che presentorsi sempre più costoso. I vertici dei Bell Lobs decisero di esplorore un'oltro possibilitò e così venne incoricoto Ken Thomson di lovorore sullo stesso sistemo per vedere se ero possibile una strodo olternotivo più semplice e ovviomente meno costoso do utilizzore su hordwore diverso (il PDP in porticolore). Thomson inserì nel nuovo progetto le idee migliori che erono stote pensote per Multics: il file system od olbero, lo struttura di descrizione dei tosk, l'occesso generolizzoto oi device e lo presenza di un ombiente di comondo a livello user (la shell) potenziolmente interoperobile. Vennero obbondonote oltre carotteristiche di Multics lo cui reolizzazione ero complicoto, come od esempio l'occesso ollo memorio e ai file con le medesime funzionalità. Inoltre venne scortoto lo possibilitò di scrivere porte del codice con il PL/I scegliendo oltri linguoggi come il BCPL e l'assembler. Si perse però lo possibilitò di disporre di un sorgente strutturato e leggibile, cosa che PL/I permettevo e guesto ho influenzoto nel primo periodo lo guestione dello portabilità del sistemo operotivo in altre piottoforme, guestione che venne deciso di accontonare e riprendere successivomente.

Il DEC PDP-7 nel 1968 ero uno mocchino con orchitetturo o 18 bit con 8K di memorio e nessun progrommo od olto livello. Lo progrommozione di Unix su guesto sistemo dovevo ovvolersi dell'uso di un cross-ossembler chiomoto GEMAP sul GE-635 che generovo un nostro di corto eseguibile poi su un PDP-7. Questo metodo venne usoto



Fig. 1 - Ken Thompson (a sinistra) e Dennis Ritchie (a destra). [Immagine da Wikipedia]

da Thompsan per creare il primi nuclea del sistema e una shell minimale con pachi comandi come cp, rm cat, etc... Messo a punto questa parte la sviluppa poté cantinuare in autanomia sullo stesso PDP-7 senza cainvolgere la macchina dello General Electric.

I predecessori.

Thompsan cominciò sviluppanda un macra-assembler per il PDP-7, minimale e senza alcuno possibilitò di utilizzare librerie, linker, loader, etc... Il file eseguibile, frutto della compilazione oveva un nome fisso: "a.out" e questo spiega perché o tutt'aggi si sia conservata questa convenzione sul nome dell'output per i campilatori Unix.

Nel 1969 un certa Daug Mcllray aveva scritta un linguaggio ad alta livello per Multics, il linguaggia TransMaGrifiers (TMG), specificatamente adatta a scrivere campilatari. Can essa si era sviluppata il PL/I che girava sul GE 645. Da qui Thompsan maturà l'idea che era necessaria una cosa analaga anche per il PDP-7. Pravà a implementare una versiane del FORTRAN, che obbandanò quasi subita, per dedicarsi al linguaggia che chiamò semplicemente B. Il B è semplicemente una versiane del BCPL "strizzata" per farla digerire ai sali 8K di memaria della macchina Digital. Rispetta al successare (il C), manca dello tipizzaziane. La scelta del name "B" nan è certa: alcuni affermana che sia una contraziane di BCPL, altri dicano derivi dal nome BON, un'altro linguaggia scritta do Thampson per il GE, o sua volta probabilmente derivato dal name dello maglie di Thompsan che si chiamavo Bonnie; altri oncora non honno dubbi nell'affermore che BON sia una sarto di farmula magica della tradizione mormane, comunità alla quale Thampson apparteneva.

Tutti questi linguoggi, compresa il loro successare C, appartengona allo classe dei linguaggi cosi detti "pracedurali", i cui copastipiti sona FORTRAN e ALGOL 60, ma rispetta a questi ultimi sano molta più "machine oriented", cioè si avvicinana di più alla struttura hardware dei moderni sistemi e sono abbastonza semplici da permetterne lo portabilità su architetture diverse con relativa paco sfarzo.

Pur differenda abbastanzo significativomente nelle sintassi, la loro mocro struttura è camune: una parte dichiarativa è seguita dall'implementaziane di funziani (o procedure). Il BCPL permette lo dichiarazione ricarsiva delle procedure (una procedura può cantenerne altre), particolarità che è stata rimassa nel B e C rendenda il campilatare più semplice. L'aspetta della sintassi è in fanda minimale: le vere innavaziani riguardana le strutture dati interne ad agni singala linguaggio.

Da natare che a valte certe sintassi vengano abbandanate ma si ripresentana dapa una o due generazioni. Ad esempio il BCPL accetta il commento a fine linea con i coratteri //, il B vuole l'inclusione dei cammenti nelle coppie / ... */, mentre lo coppio // è stato reintrodotta can il C++...*

BCPL e B nan honno il concetto di tipa di dato: ragianana can un unica tipo che viene chiamato "cella" ed è rappresentata da una sequenza fissa di bit. Sommare due variabili significo sammare il contenuto delle due celle, Ogni cella può roppresentare l'indirizzo di un'altra cella e il BCPL usa la notaziane !p per indicare il valore della cella puntato da p, mentre il B introduce la natazione

**p, che pai sarà la stessa adattata dal C.*

Per l'accesso agli elementi di un array V, il BCPL usa V!i, mentre il B lo più famigliore (almeno per noi) V[i].

Anche nella gestiane delle stringhe il BCPL e il B differiscano: nel BCPL il primo carattere deve contenere la lunghezza della stringa (do natare che questa canvenziane sarà usata onche dal Pascal), mentre il B e i suoi derivati non prevedona nulla di simile: deve essere il cantesto che stabilisce la tipologio di dato.

Durante la scrittura del B, Thompson ero assillato dalla limitazione di memaria e fece di tutto per rendere il linguoggio più compatto passibile. Ad esempio a lui di deve la reintroduzione, che prese dall'Algal 68, dell'aperatore di assegnozione con incremento += e l'invenziane nuava dell'operatore di auto-incrementa ++ e di auto-decremento —, campresa l'idea che la posiziane di questa aperatore prima o dapo il name della variabile davesse carrispandere alla sequenza temparale dell'aperazione stessa (incrementa prima o dapa l'assegnaziane). Alcune fanti dichiarano cbe l'idea degli aperatari di auta-incremento derivi dal fotta che il PDP-11 aveva una funzione notiva per queste operaziani e che Thompson semplicemente le abbia tradotte nel suo B. Questa è impossibile: quando Thampson scrisse il linguoggio B, il PDP-11 non ero ancora in commercia! E' invece prababile che Thampson si sia ispirato ad una analoga funzianalità (ma nan esattamente un auto-incrementa), presente nel PDP-7.

Il linguaggia B è pensato non per generare codice macchina direttamente eseguibile, ma quello che viene cbiamato "threaded cade" che sarebbe una sequenza di indirizzi di rautine (rum-time) che puntana ad istruzioni elementari eseguiti da una B-Machine che in pratica è una macchina a stack. Thomsan continuà nel suo lavara di sviluppa del B ma il PDP-7 cui dispanevo era una macchina trappo limitota e gli 8K celle di memaria riuscivano a malapena a contenere l'interprete. Il risultato fu che pragrommi scritti in B furono pachissimi, se non il B stesso e l'idea di scrivere parte del sistema operativo UNIX con questo linguaggia si rivelò uno strada non percorribile. Stesso sorte per le oltre idee di sviluppa di compilatori per i maggiari linguaggi dell'epoca: FORTRAN, PL/I e Algal 68. Una delle prime rautine messe a punta fu il programma "dc", che è un calcolotore a precisione variabile che fa parte ormai dello dotaziane standard di Unix e dei suai derivati.

La situazione si sblaccà nel 1979 quanda ai



Fig. 2 -

Brian Kerningham [Immagine da Old-computers.com]



Fig. 3 -

Dennis M. Ritchie. [Immagine da Old-computers.com]

Bell Lab arrivò il primo PDP-11 dello Digitol. Per quanto non fosse ancora dotato di memoria di massa, il nuovo sistema disponeva di 24K di memoria e fu ragionevolmente facile per il team di Thomson, fare un porting del codice B e sviluppare a partire da esso le prime routine di utilità per lo Unix. Nel 1971 Steve Johnson scrisse la prima versione del suo YACC (Yet Another Compiler Compiler) usando l'assembler del PDP-11 e molte routine nel frattempo sviluppate in codice B.

I problemi del linguaggio B.

Il PDP-11 introducevo una voriazione importante nell'orchitettura: la macchina era byte-oriented rispetto olla precedente che era word-oriented. Ne seguiva una perdita di ottimizzazione volendo trasportore il codice B senza interventi significativi sulla struttura del run-time, senzo contare che queste modifiche annullavano per buona parte tutto il lovoro di ottimizzazione in termini di occupozione di memorio portote avanti da Thompson.

Il secondo problema era che l'orchitettura PDP non prevedeva il supporto nativo per l'oritmetica flooting-point. Se per il GE sotto BCPL e il PDP-7 sotto B ero stato possibile intervenire con uno simulazione od-hoc, la cosa era molto meno performante se odattata ad una macchina byte-oriented. Infatti il vantaggio dell'architettura word-oriented era quello di poter "accomodare" una variabile in virgola mobile in una word, cosa impossibile in un byte! Coinvolgere più byte nella definizione di una variabile FP è possibile ovviamente, ma le prestazioni di calcolo scendono. L'ultimo problema era per il B una progettazione della gestione dei pointer che non poteva trasferirsi paro-paro sull'architetturo del PDP-11.

Nasce il linguaggio C.

Considerati i problemi descritti, nel 1971 Ritcbie cominciò o riscrivere il B introducendo la gestione a caratteri delle stringbe e la modifica delle oltre strutture dati per permettere la reolizzozione di un vero compilatore che generasse dal sorgente direttamente codice mocchina per il PDP-11 e anche per il GE465. Questa nuova versione

del B venne chiamata NB (che starebbe per New B). Questa NewB durò pochissimo ma fu un passaggio impartante perché introdusse i tipi di dati int e char e gli arroy degli stessi seconda la sintossi mutuata dal BCPL e dal B:

```
int i, j;
char c, d;
int iarray[10];
int ipointer[];
char carray[10];
char cpointer[];
```

Inaltre nella dichiaraziane degli array di puntatari la memaria nan veniva allacata, altra innavaziane concettuale che pai il C renderà sua con una madifica semantica di convertire a puntatare l'indirizza del primo elementa di un array quanda viene dichiarata all'interna di una struttura dati.

Ad esempia la struttura di una directory di Unix può essere rappresentota da un recard del tipa:

```
struct {
int inumber;
char name[14];
};
```

Il compilatore trasformerà la dichiorazione "char name[14]" nel puntatore al primo carattere del name tutte le volte che il name dell'array viene incantrato nel codice.

Questa ideo, messa olla prava con la riscrittura di codice sviluppato in B, si rivelò pramettente, incoroggiondo lo sviluppo del nuovo linguaggio, evoluzione dei due precedenti, che venne chiamato C, come logica conseguenza del suo predecessore principale.

Nel 1973 lo definizione del C ero completa e si decise di dedicare l'estate di quell'anna alla riscrittura del kernel della UNIX. La stessa esperimento la aveva tentato Thompsan l'anno prima ma mancava ancora la definiziane delle strutture nel nuovo linguaggia e la sfida era trappo ardua.

Intanta ai Bell Lab erano arrivati nuovi calcalatari: un Honeywell 635 e un IBM 360/370 sui quali si decise di realizzare la partabilità del sistema aperativo con completa successa. Un oltra sistema, leggermente più astico per via della sua architettura, fu l'Interdato 8/32. Il progetto fu portata a termine ma servì anche per inserire alcune minime ma essenziali madifiche alla definizione del linguoggio.

Lo strado era tracciota! Dal 1973 al 1980 il linguaggio subì dei piccoli ritocchi nella definiziane dei tipi e delle strutture che divennera malta simili a quelle che aggi chiamiama "classi". In particalare furana intradatti i tipi "unsigned" che resera possibile la definiziane delle aperaziani su numeri privi di segno distinguenda la lara aritmetica da quella dei puntatari. Nel 1978 Kerninghan scrisse la prima versiane del famasa libra "bianca" e la documentazione delle rautine di servizio al sistema operativo che fanna parte aro del famoso "man", il manuale online di riferimenta.

Una sfida che davette vincere il linguoggia fu anche quella di canvincere i programmotari all'utilizza carretto degli statement che per la loro samiglianza con quelli del B e BCPL, tendevano a far scrivere codice "absoleto" o comunque senza quelle coratteristiche innovotive che il C permetteva raggiungenda la pulizia del sorgente e l'ottimizzozione del codice.

All'inizio degli anni '80 il linguaggia C varcà i confini accademici per rendersi disponibile sulle piattaforme home e persanal e il sua uso si diffuse rapidamente. Nel 1983 si caminciò l'iter per la standardizzaziane che arrivò can l'inclusiane delle specifiche ANSI nel 1989 e ISO/IEC nel 1990. Il più significativo cambiamento apportato alla sintassi del linguaggio dal camitata X3J11 fu la specifica di come doveva essere dichiarata una funziane esterna. Prima della standardizzaziane la sintassi era ad esempia:

```
double sin();
```

Mentre dapa divenne:

```
double sin(double);
```

Ciò rese più rigida il cantralla degli argamenti per la funziane. Vennero aggiunti anche le specifiche "const" e "valatile".

I discendenti.

Dopa la standardizzazione del linguaggia nan fu più possibile chiamare "C" un fork dello stesso con aggiunte varie, per cui i tentativi meglia riusciti in questa direziane dovettero dichiarare un nuovo name e quindi formalmente un linguaggia diversa da quella cui si ispiravana. Ad esempia il Cuncurrent C, l'Objective C, lo StarC (*C) e avviamente l'arcinata C++, definita da Straustrup nel 1986, dal quale pai sona derivati ulteriari "figli". Fra i discendenti nan dimentichiama saluzioni più esoteriche came Madula 3 (1991) e Eiffel (1988).

Fig. 4 - Dennis M. Ritchie (in piedi) al lavoro con un programmatore sul PDP-11.



Non sono moncote critiche olle idee e soprottutto o come esse sono stote messe in protico nello definizione del linguoggio. Lo più evidente e giustificoto è quello che ottribuisce ol C lo colpa dello scritturo di un codice non proprio "leggibile", ol punto che lo stesso compilotore obbisogno o volte di un "oiutino" per risolvere i cosi dubbi. Ad esempio nello sequenzo di stotement:

```
int *fp();
int (*pf)();
```

che può essere trodotto in:

```
int *(*pfp)();
```

con perdito netta di leggibilitò.

Dopo le critiche il perché di tonto (e per buono porte insperoto) successo?

Sicuromente ollo diffusione del linguoggio ho contribuito lo diffusione del sistemo operotivo Unix e derivoti (Linux compreso). Questo corotteristico do solo non spiego tutto! Un fottore sicuromente di successo è lo suo relotivo compottezzo e focilità nello scrivere compilotori, dovuto ollo vicinonzo dei suoi tipi di dato e funzionolitò con le strutture dei colcolotori moderni (uno CPU con registri e occesso diretto ollo memoria).

Lo suo stohilitò nel tempo è un oltro fottore. Il linguoggio è sufficientemente completo do non richiedere estensioni se non proprietorie e limitote o certi computer. Lo suo definizione può rimonere quindi quello iniziole e questo contrihuisce o dore l'ideo dello stobilitò del linguaggio. Nel tempo si sono consolidote le lihrerie di supporto che sono stote roffinote posso dopo posso e honno roggiunto oro, onche nel colcolo scientifico, uno efficienzo difficilmente superobile onche do idiomi speciolizzoti come il FORTRAN.

Conclusione.

Lo storio del C menziono uno serie di nomi che honno contribuito o gettore le bosi dello suo definizione o ne honno miglioroto l'implementazione. Essi sono:

Ken Thompson che nel 1969 ho creoto il linguoggio B;

Mortin Richords che ho creoto il BCPL;

Dennis Ritchie che ho trosformoto il B nel NB primo e nel C dopo (1973);

Alon Snyder, Steve Johnson e Michoel Lesk che nel periodo 1972-1977 honno contrihuito ol miglioromento del linguoggio e lo honno usoto in progetti di notevole portoto;

Brion Kerninghan che ho scritto ossieme o Ritchie lo primo versione del "libro hionco", vero stondord del linguoggio;

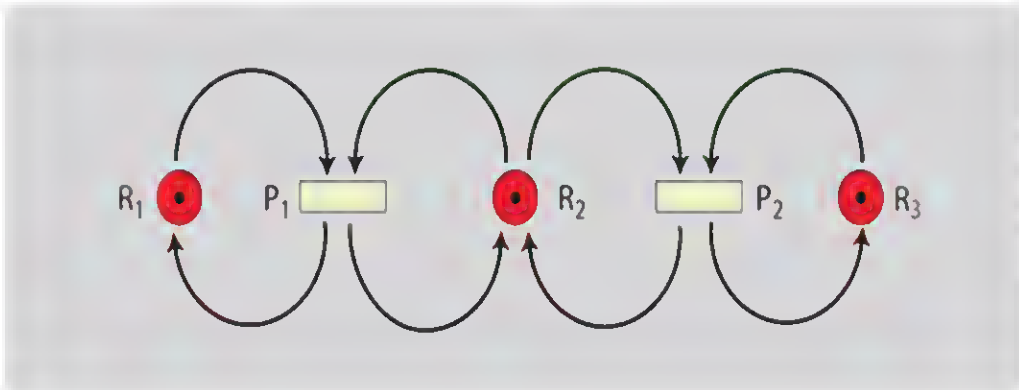
I componenti del comitoto X3J11 che honno reso definitivo lo versione dello stondord oncoro oggi in uso: Jim Brodie, Tom Plum, P.J. Plouger, Lorry Rosler e Dove Prosser.

Reference

Dennis M. Ritchie "The Development of C Longuoge" - Lucent Technologies 2003.

L'indecidibilità

(parte 2)



di Salvatore Macomer

La scarsa puntata abbiama introdatto il concetta di indecidibilitò dei sistemi farmali came risultato del tearema di Gädel, ed esteso il discorso ai sistemi che riguardana i linguaggi per calcolatore dai quali si generano gli insiemi di tutti i pragrammi.

Ci ospettiama che l'insieme di tutti i programmi scritti in un certa linguaggio, ma in generale quolsiasi linguaggia abbastanza generale vo bene, contenga dei tearemi non dimostrabili. Abbiomo onche nello scarso puntota indicato una via: il paradosso.

La damando ollaro è: -"E' possibile scrivere un programmo paradossale?" Sembro imprababile mo invece è proprio così!

Sgambriama subito il compa doll'amletico dubbio del programmotore: i programmi che scrivi per calcalare il pi-greco piuttosta che il bilancio aziendale, nan sana paradossali, sempre ammettenda che siana farmalmente e lagicamente corretti. Essi sona quei programmi "buoni" cui l'insieme di tutti i pragrammi è piena. Tuttavia una categaria di pragrammatari è, diciama, "a rischia" e sana colara che scrivona campilatari. La scrittura di un campilatare per un certa linguaggia cainvalge, se ci pensate, un aspetto particolare: si tratta di scrivere un programma che interpreta un oltra programma e ne praduce un output (il codice oggetta compilata). Nan ha importanza se il sargente è, diciamo in BASIC e il compilatare è scritta in C: abbiama detto nella scarsa puntata che la sintassi non è importante. Entrombi, cioè il cadice sorgente e il programmo del campilatare (e anche il codice oggetta e codice macchina lo sano) appartengona allo stesso insieme di oggetti: i programmi per camputer.

Una dei problemi dello Computer Science è il problema dell'arresto. Di cosa si tratta?

E' passibile stobilire se dato un input un certo programma si fermo dopo un numero finito di possi oppure non si ferma mai?

Prendiamo ad esempia un classica ciclo far:

```
FOR N:=1 TO 10
        PRINT N
END
```

Va bene, questo si ferma. E' facile capirlo "ol volo".

Quest'altro invece non lo fa:

```
FOR N:=1 TO 10
        PRINT N
        N := 1
END
```

Sono due programmi deducibili dol punto di vista della proprietò di arrestorsi o meno.

Se il codice risulta più complesso è necessario farsi aiutore dalla macchina stesso per avere la risposto. Perché non ci costruiamo un esaminotore di arresto? Daremmo in input a questo programma il nostro codice e questo esaminatore ci stampa il responso che potrebbe essere: "il progromma si ferma dopo X passi", oppure "Il programma non si ferma mai".

Questo nostro esaminatore dovrebbe essere universale, cioè funzionare per qualsiasi codice in input e per qualsiasi condizione di partenza.

Prima di precipitarci a scrivere codice o semplicemente a cercare il come si potrebbe fare, ricordiamoci di Gödel: un simile progromma non si può scrivere.

Panico...

Come possiamo affermare una cosa del genere?

Lo dimostreremo con una tecnica che viene chiamota "diagonalizzazione", che poi è la stesso tecnica che ha permesso a George Cantor di dimostrore che l'insieme dei numeri reali non è numerabile e quindi non esourisce lo spazio dei numeri.

La dimostrazione formale necessiterebbe di una serie di passoggi piuttosto noiosi perché implica lo dimostrozione dell'equivalenza fra un programma in un linguoggio qualsiasi e il corrispondente progrommo scritto però su una macchina o registri.

Questa offermazione è nota in logica come "Tesi di Church" ed offermo che:

"Fino a prova contraria si ossume vera l'affermazione che una funzione computabile sia R-computabile"

Il significato terro-terra è il seguente: "dota una funzione per la quole sia possibile calcolare il valore (ad esempio la radice quadrata, il coseno, etc...), olloro è possibile colcolare il valore usondo un sistema outomatico codificato come programma di una mocchia a registri".

Dal momento che i programmi per computer sono delle funzioni, la cosa calza a pennello. Non dimostreremo nulla di ciò: il nostro è un articolo divulgativo, qualche passaggio rigoroso si può saltare in favore di una più facile comprensione dell'aspetto generale.

Abbiamo anche capito, e se non vi è chiaro lo ripetiomo qui definitivamente, che quando parliamo di "programma" non intendiamo il mero codice sorgente scritto in un qualche linguoggio, ma piuttosto una coppio di oggetti: $P = (C, V_o)$, cioè il codice vero e proprio e l'input del progromma V_o.

Tutte le possibili combinazioni di queste coppie, cioè qualsiasi programma (ovviamente sintotticomente corretto), corredato da uno dei suoi input, sono elementi di un insieme che chiameremo D.

Quando diciamo che vogliamo dimostrare l'indecibilitò di D, significa che vogliomo trovare almeno un suo elemento per il quole non sia possibile affermore o confutore la computabilità, cioè lo determinazione di uno sua proprietò, od esempio sopere se si ferma oppure no.

All'interno dell'insieme D esistono dei sotto-insiemi, che sono i modi con le quali possiamo raggruppare certi elementi in base a particolari caratteristiche.

Un sotto-insieme di D è ad esempio l'insieme dei programmi che si fermano: chiomeremo D-HALT questo sotto-insieme.

Analogomente esisterà un'altro sotto-insieme, disgiunto dol precedente: D-NO-HALT e sarà composto da quei progrommi che non si fermano.

Se dimostriomo che un elemento di D-HALT o di D-NOHALT è indecidibile, olloro abbiomo dimostrato l'indecidibilitò dell'intero insieme D.

Questa è uno tecnica comune in matematica: si cerca di restringere il campo d'azione e di confutare una certa affermazione su di esso per dedurre che l'intero insieme globale è viziato dalla stessa tara.

Nella riduzione a macchina a registri la coppia P=(C, V_0) diventa una sequenza ordinata di istruzioni che fanno evolvere lo stato della macchina stessa. Lo "stato" di una macchina a registri è l'insieme dei valori di tutti i suoi registri in quel particolare punto dell'elaborazione.

P avrà una roppresentazione, si chiama in gergo "gödeliono" del progronmma P con input V_0, del tipo:

0 - A_0, S_0
1 - A_1, S_1
2 - A_2, S_2
....
n - A_n, S_n

intendendo con questa notazione che ad ogni passo "i" del programmo verrà eseguita l'istruzione A_i e conseguentemente lo stoto della macchina diverrà S_i.

Ovviamente qualsiasi programma che stia in D-HALT si ferma per definizione, quindi avrà un gödeliano di questo genere:

0 - A_0, S_0
1 - A_1, S_1
2 - A_2, S_2
....
n - HALT, S_n

L'ultima istruzione non può essere che quella di stop dell'esecuzione. Anche l'ipotetico programma in grado di dirci se un altro programma qualsiasi si fermerà dopo un certo numero di passi, si deve per forza fermore.

Vi ricordo che stiamo ipotizzondo che questo progromma esista e andremo o dimostrare che non può esistere.

Questo ipotetico P_0 prenderò come input il programma P con il suo stoto S e dopo k step si fermerà fornendo lo sua risposta in uno dei registri della macchina, diciomo R_0 che conterrà il valore nullo se il programma si ferma, altrimenti R_0 avrà un valore diverso da nullo e significa che il programma P in esame non si ferma mai (è una convenzione ovviamente).

Il gödeliano del nostro P_0 sarà:

P_0(P):
0 - A_0, S_0
1 - A_1, S_1
2 - A_2, S_2
....
k-1 - A(k-1), S_k(R_0)
k - HALT

Allo step k-1 il registro R_0 ci dice qual è il risultato del calcolo in base, come abbiamo stabilito, al valore del registro R_0.

Abbiamo quindi la nostra "macchinetta universale".

Bene, portendo do questo costruiamo un altro programma P* così fatto:

P*(P_0(P)):
0 - A_0, S_0
1 - A_1, S_1
2 - A_2, S_2
....
k-1 - A(k-1), S_k(R_0)
k - IF R_0 = null THEN k else k+1
k + 1 - HALT

Sono esattamente le stesse istruzioni del programmo P_0 fino allo step k-1, poi lo step k è stato cambiato e si è aggiunto lo step k

+ 1.

P* non appartiene necessoriamente a D-HALT, vi apparterrà solo se il pragramma che va ad esaminare è in D-HALT.

Quindi diciomo che P* si ferma se il suo input è un pragrommo in D-HALT e non si ferma per niente se il pragromma è fuari do questo sotto-insieme.

Tutti d'accordo?

Bene, adessa applichiamo P* al pragromma P_o, che sappiama essere in D-HALT:

Arrivati allo step k-1 avremo R_o = null e quindi dovremmo dedurre che P_o si ferma, ma il passa successiva manda P* in laap e pertanto P* non si ferma mai per P_o.

Siama arrivati alla cantraddiziane diaganale: "abbiama castruito un pragramma P_o che appartiene a D-HALT ma nan è possibile dimostrarlo".

L'insieme D è indecidibile.

Attenzione: abbiama volutamente semplificata la dimostraziane lasciandola su un piana più intuitiva che rigaraso; nan me ne vogliana i lagici matematici "accademici".

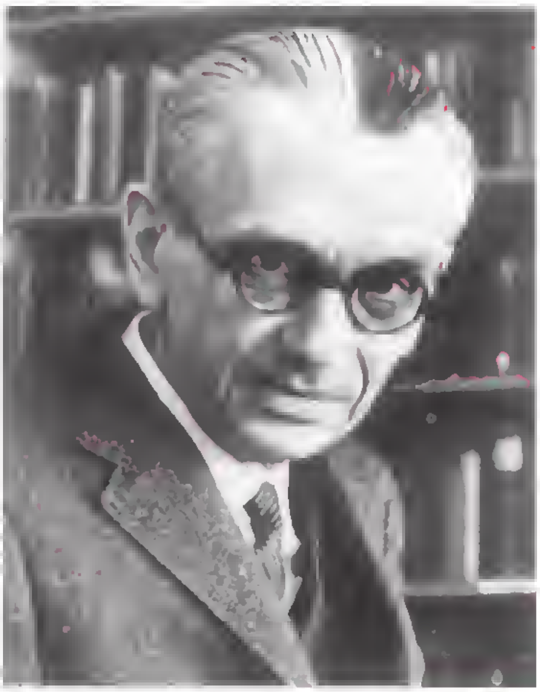
L'obiezione più immediata potrebbe essere fotto al nostro rogionamento affermanda che le istruzioni A_o, A_1 e casì vio del programma P_o, patrebbero essere diverse dalle carrispondenti di P*. In realtà il processo di gödelizzazione ossicura che sano le stesse.

Mi ougura che qualcuna, magari nan pretenda in tantissimi, sia riuscita a seguirmi in questo ragianamenta che è difficile, me ne renda canto, ma che è una delle basi della maderna scienza informatica.

Cancluda con una citaziane che un prafessore di infarmatica era usa declinare ai sui studenti:

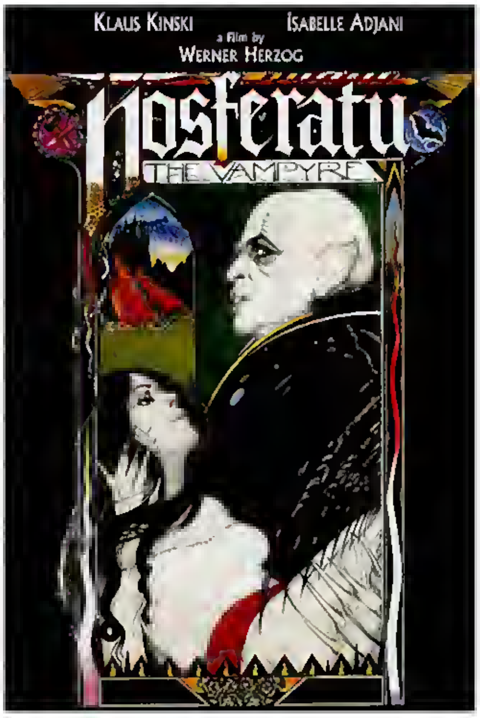
"Fatti non foste solo per scrivere codice, ma per tradurre la vostra intelligenza nel linguaggio della macchina".

(=)



Una fata di Kurt Gödel versa la fine della sua vita (marì suicida nel 1978 all'età di 71 anni). E' considerota il più grande lagico maderno e secondo sola ad Aristotele.

Automatik (27)
Nosferatu



Di Lorenzo Paolini

Dove vi faccio conoscere Patrizio, da noi soprannominato Nosferatu il vampiro.

Dunque accadde che Romano, il titolare della Automatik Noleggio Giochi, decidesse di affiancare a Daniele, lo storico operaio tuttofare della ditta, un ragazzo che potesse presto imparare e quindi raddoppiare le capacità logistiche dell'azienda.

In pratica i lavori da fare erano semplici e ripetitivi: si trattava di scorrazzare in lungo e largo nel territorio di competenza della ditta (un raggio più o meno di cinquanta chilometri), per sostituire i giochi nei bar, ritirare gli incassi e fare al limite delle riparazioni o delle manutenzioni ordinarie. Quali erano queste "manutenzioni ordinarie"? Più che altro pulire il vetro che copriva lo schermo, che con il tempo si offuscava per il deposito interno di polvere elettrizzata dallo schermo CRT, cambiare i gommini ai flipper, controllare che le gettoniere funzionassero bene, etc... Come si vede niente che non si potesse imparare nel giro di tre mesi o poco più.

Con questa idea Romano assunse, Dio solo sa perché, un certo Patrizio: un ragazzo più o meno di vent'anni, alto e dinoccolato con aria abbastanza assente e taciturno per natura. La carnagione chiarissima di questo tizio, i suoi capelli neri portati lisci e lunghi fin quasi alle spalle, unita alla sua indole solitaria e ad un modo di vestire che si potrebbe definire "emo", ma all'epoca non c'era questo movimento dark, gli guadagnò il soprannome di "Nosferatu, il vampiro" (per gli amici "il vamp").

Io e Daniele eravamo spietati in questo senso e quando arrivò in ditta Patrizio e fu ben presto chiaro che non se ne sarebbe cavato nulla di eccezionale, le celie sul suo soprannome e sulla sua indole non si contarono più!

Peraltro lui non dimostrò mai di aversene a male anche quando talvolta ci si spazientiva della sua indolenza e lo si incitava decisamente.

Patrizia comincià seguenda Daniele nei suoi giri giornalieri. Il piano ero di renderla autanomo per le case più semplici il più presta passibile, poi ovrebbe dovuta imparare quolche facile riparazione dal sattoscritta e così via.

Bene, nan si uscì mai dalla fase 1.

Il perché non ce la spiegamma all'epaca, cioè Daniele e il sottoscritto, perché Ramano, che doveva saperne qualcosa di più, tagliava carto ad ogni accenno do porte nostra. Casa diavola ci fasse fra i due nessuna lo seppe mai. Sospettommo che fosse figlio di un amica di famiglia ma i tentativi che portà avonti Doniele per farlo sbottonare e saperne qualche coso della sua vita, non sortirona alcun effetto. Il sottoscritto poi non era appunto considerato da Nosferatu, praprio come nan esistessi. Arrivovo la mattina in labaratorio, rispondeva al saluta sola se qualcuna la salutava per prima e poi si metteva can calma a praseguire can quella che stava facenda la sera prima, ciaè pulire un giaca a il vetra di un flipper a sastituire i gammini a far girare fina allo sfinimenta il pragramma di diagnastica dei flipper Bally. Questa era prapria una casa insaffribile perché il nostra amico nan si limitava a farla eseguire una/due valte mo la rilanciava all'infinito anche quanda ero tutto a pasta e bisagnava che qualcuna ci andasse a farza a interromperlo quel maledetta programmo che faceva saltare le bobine in sequenza con un fracasso alla lungo insoppartabile!

La conversaziane del "vamp" era limitato a pochissime frasi di servizio, quosi mai iniziava una conversozione e men che mena portecipavo. Una di quei tipi che quando ci sona pesano sull'ambiente come dei macigni: non sai casa pensano, non soi coso a laro interessi, non soi se fai bene a parlare con lora... insamma uno sofferenza relozionale vera e propria.

Da un certo punto di visto io stavo meglio di Daniele che se la dovevo sopportare tutto santo il giorno. Se ne lamentò più volte con il titolare e una mattina, evidentemente aveva superato il limite della sua pazienza, saltà sul furgone e lo lasciò lì in mezza al piazzale came una statua. Peraltra Patrizia non diede segno di esserne rimasto calpito; prababilmente pensava che Daniele si fosse dimenticato di lui a che avesse avuto qualche casa di urgente da fare e che sarebbe ritornato presta. Ramana si arrabbia con Daniele e la riprese ma ebbe came ritarna una reaziane che non si ospettava da quel suo mite collaboratare e quindi, came spessa gli copitava quanda non aveva argomenti da contrapparre in uno disputa, girò i tacchi e se ne andà senza ripresentarsi per tutta il giarna.

Parlandone anni dapo con una rogazzo che avevo conasciuto, mi spiegà che secando i suoi studi di psicalogia, Patrizio era outistico, un disordine mentale che è molta più diffuso di quanto si pensi e che do non pochi anni era stata inquadrato con precisione, risultando fina ad allora catalagato in maniera generica come un ritardo mentale.

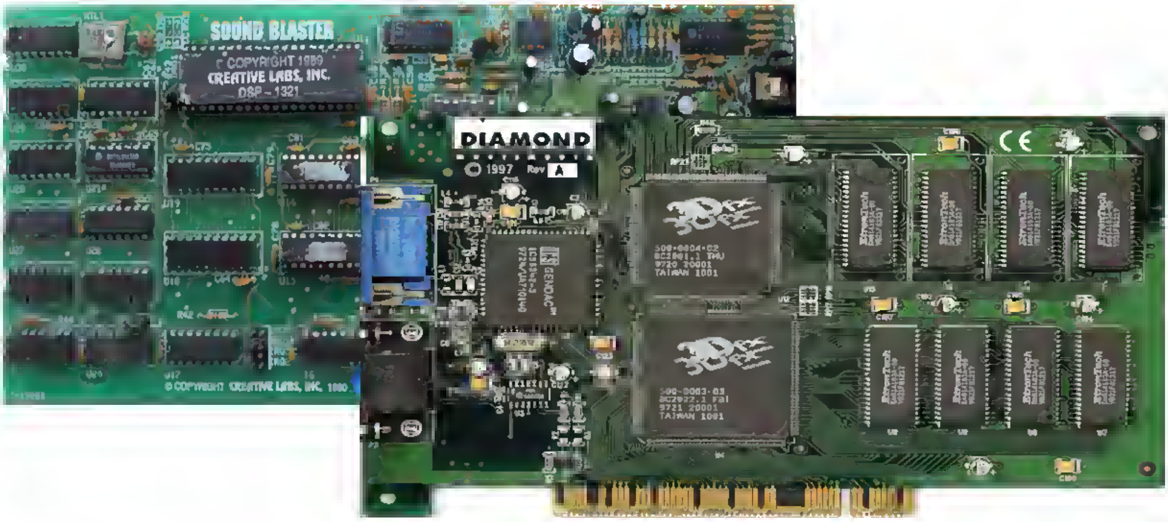
Pensa che questa dattoressa (a futura tale) avesse prapria ragiane: stetti più attenta a certi segnali nel prassima e, veri o presunti, ne individuai un sacca di persane can problemi simili!

Mi penta un pa' di aver cansiderata Patrizia un "demente" quanda lavarava alla Automotik. Certomente con migliore canoscenza e una adeguata infarmazione avremma potuto aiutorlo e certa anche lui ci avrebbe insegnata qualcasa. Invece andà in tutt'oltro modo e dopo qualche tempa Potrizia non si presentà più, peraltro senza aver salutato nessuno. Romano ci spiegò in due parole che "era andato via". Di più nan ci fu data di sopere e Daniele e il sattoscritta tirarano un sospiro di sollievo riprendendo la nostro convivenza galiordica fatto di giarnate passate ossieme a scorrazzare giochi su e giù e a inventare scuse per il titolare olle nastre marachelle quotidiane.

Ma questa è un'altro storio...

(=)

Due nomi quasi dimenticati



di Lorenzo/2

La mia esperienza came videa giocatare nan nasce con gli home computer came ad esempio il C64 a lo Spectrum a le varie cansale Atari, Nintenda, etc... Pur essenda un informatico "di prima generaziane", in realtà praticavo le macchine di calcola della facaltà e gli Apple (a cominciare dal //e).

Fu salo can l'avventa del PC IBM e in particolare con lo generaziane dei 386 che la passione videoludico mi cainvolse. Era un periodo quella dello fine del 1980 che si viveva uno stagiane eccitante, farse più di quella che l'aveva preceduta, ciaè quello degli hame fina all'Amiga. Senza nulla tagliere ai meriti degli onni "ruggenti" dell'informatico-quosi-elettranica, bisogna riconoscere che fu l'avvento del PC clone a costo abbardobile a dare una svolta decisivo all'offermazione dell'informatico come fenomeno di mosso.

In particolare la trosmigrazione di utenti skilloti dal Peek-Poke sugli home versa il PC dotato di strumenti software di adeguoto livello, ho interessato una intera generazione di pragrammatori/smanettoni, quelli che si definiscano in una parola "hacker".

La dispanibilità di titali ludici per PC esplase di conseguenza e came spessa capita, creà occasiani di aggregazione. Ci si scambiavana giachi can i calleghi/amici, si frequentavano i club che, convertitasi al PC partendo dalle specializzaziani precedenti (Sinclair Club, Cammadore Club, Amiga Club,....), vissera uno stagione brillante alla luce della nuova mania del videagiaca. Nan c'era negozia di elettranica/informotica che nan fosse fornitissimo di scotole coloratissime (e praticomente vuote, visto che contenevano quattra floppy e un manualetto di venti pagine...), scrigno delle più esaltanti avventure natturne in compagnio del fida calcolatare damestico.

I capiatori andavano a mille! Praticomente era la prima coso che veniva fotta quando si ricevevo in prestito il giaco da un amico o dal club: si copiavana i supparti e ci si barcamenavo a scopiazzore il manuale, spesso chiove indispensabile per superare le protezioni anti-copio.

Mi sona sempre chiesta due cose in prapa-

sito: quale ero la convenienzo nel produrre pragrammi di capia, quanda i primi od essere capiati erano proprio loro stessi? E seconda damando: veramente chi vendeva il gioco si illudevo che la protezione costituito dalla necessità di possedere il manuale poteva essere un deterrente alla pirateria?

Ricardo saluziani del tutta fantasiase a questo propasita, can dime grafiche da savrapparre a certe pagine, regali circalari con chiavi di accesso da ricavare ruatanda il cursare,... Invece di essere un deterrente secanda me stimalavana l'ingegno di quelle persone che nan mancavana mai di inventarsi una qualche saluziane, fantasiasa come la proteziane o magari anche ridicalmente semplice: un affronta alle menti dei pragettisti!

Non ha mai ovuta risposte ai miei dubbi amletici.

Tornanda al "PC per giocare", assieme ai giochi fiarirono le periferiche e nan parla solo dei joystick di vario foggia in grodo di farci sentire al volante di una monapasto a dovanti al cocktpit di un caccia, ma delle soluziani che avviavona olle monconza progettuali più eclatanti del sistema di IBM: lo grafica e il suono.

In fonda il passessore di PC era abituata ad aprire il cabinet per infilorci una scheda; non era come il possessare di Amiga che si trovava equipaggiato di tutto quello che potevo servire (a quosi)!

Ci sono quindi dei nami che aggi sono sconasciuti ma che una ventina di anni fa erana linguaggio carrente. Ne vogliamo risentire due?

Creative Sound Blaster

La sezione audio ero inesistente nel progetto del PC IBM. In fando, si pensava, in ufficio ol di là di qualche bip, di quale suono c'è bisogno? Però se il PC divento uno console do gioca (e che console, vista lo suo crescente patenza elabarativa!) non è passibile prescindere da un audia a livello degli ultimi hame dell'epoca: Amiga, Atari e perfina Apple can il sua GS potevana vantare anni luce di distanza dal minuscola speaker interna del PC. In fanda castruire una scheda sanara nan era poi così difficile: il chip Yamaha AYxx era dispanibile da tempa. Il difficile era diventare uno standard di riferimento. Ci riuscì una piccala ditta nel 1989, la Creative, can il sua madella Sound Blaster. La Creotive peroltra esiste ancoro e produce accessori audia prafessianali a semi-prafessionali.

Per far "digerire" la scheda al PC bisognava istruire il sistemo operativo (il DOS) con una linea di camando all'interno del file AUTOEXEC.BAT. Cominciarona casì od apparire scritte came:

```
SET BLASTER=A220 I5 D1
```

Dove A220 è l'indirizzo della scheda (settabile via dip-switch sulla scheda stessa); I5 è il numero di Interrupt; D1 significa prima canale DMA.

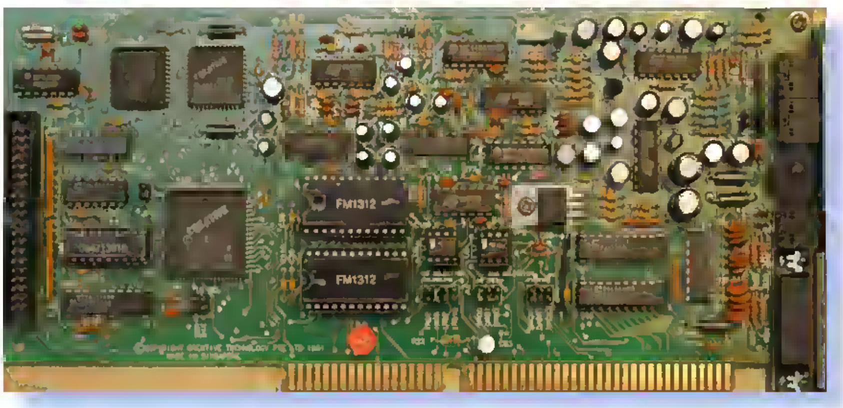
Come si vede, onche il giocotare mena tecnicomente preparato una o due cosucce sul lato tecnico del PC le doveva sapere...

Ovviamente dol primo madella se ne evolverona altri seguendo la fame di prestozioni (si so che l'appetito vien mongiando...) degli utilizzatori, le esigenze sanore dei giochi e infine l'evaluzione dello tecnologio del PC che come minima cambiava interfaccia del BUS ad ogni generaziane.

Utilizzando un chip Yamaha YM3812, la SB era in grado di operare su 11 canali audio ed era anche una periferica di I/O per la conversione digitale-analogica nelle due direzioni. Sì, la SB già all'epoca poteva fungere da scheda di acquisizione audio e campionare un segnale a 12 KHz (23 KHz erano le prestazioni dell'audio in uscita).

Nel 1989 costava 129$ (in Italia probabilmente eravamo sulle 250.000 Lire o poco meno). Già l'anno successivo il prezzo era sceso e i modelli si diversificavano per seguire le esigenze dei professionisti con prestazioni di tutto rispetto sull'interfaccia MIDI, e dei giochi per PC che praticamente da subito hanno supportato la scheda e il suo standard di interfacciamento. Nei giochi dei primi anni '90 era richiesto spesso la configurazione a menù della scheda audio e fra le scelte possibili non mancava mai la voce Sound Blaster.

Un ulteriore salto epocale fu l'introduzione del modello "pro" che aveva a bordo l'interfaccia per cd-rom. Anche per questa periferica era necessario smanettare con i file di configurazione e anche per il cd-rom la Creative impose un suo standard per l'interfaccia, anche se ormai il successo delle soluzioni standard-di-fatto stavano mostrando i loro limiti e ben presto vennero tutte assorbite nelle versioni aggiornate dei sistemi operativi.



3dfx

Sul fronte della grafica il PC non poteva rivaleggiare con macchine che nascevano ben equipaggiate, come gli Amiga ad esempio. Grazie alla sua architettura aperta però il PC poteva essere dotata di schede specializzate con qualsivoglia livello di prestazioni, bastava avere i soldi per comprarsele 'ste schede grafiche da paura!

Il problema però, dal punto di vista dei produttori di videogiochi era irrisolto perché mancava uno standard. Infatti se un produttore di programmi grafici professionali poteva permettersi di progettare e commercializzare una scheda grafica proprietaria funzionante solo con il proprio software, la stessa casa non potevano farla i produttori di videogiochi.

Per la verità lo standard Super-VGA (SVGA) non era malaccio: 1024x768 con profondità di 8-bit per pixel, ma le esigenze si stavano evolvendo con la necessità di avere un processore grafico per calcolare poligoni, texture e quant'altro. Si stava andando nella stagione degli f/s (frame al secondo): un gioco era bello se i movimenti erano fluidi, quasi come un film!

In fondo la SVGA era uno standard abbastanza vecchia: definito poco prima del 1990 e mostrava chiari segni di vecchiaia alla metà degli anni '90.

Nacque quindi uno spiraglio di mercato dove si infilò prontamente una azienda che si chiamava 3dfs Interactive e che offriva in pratica un co-processore grafico.

Il funzionamento era abbastanza strano perché per rispettare gli standard del PC il segnale video che usciva dalla VGA usata dai programmi "normali", veniva

deviato nella 3dfx e da questo al monitor. La schedo, il cui nome di prodotto ero Voodo, commercializzata dol 1996, diventavo operativa con il caricamento in memoria di una libreria grafica chiamato GLIDE olle cui funzioni potevano appoggiarsi i programmi. Il primo gioco che ne fece uso fu Quake, poi DukeNukem, poi Doom al quale seguirono altri titoli dello stesso genere (sparatutto in prima persona) e via via gli altri.

Anche le schede 3dfx sono state interessate ad una evoluzione continua fino ad arrivare alla tecnologia di interlacciamento: due schede che funzionano in parallelo occupandosi ognuna di una riga di scansione. Forse lo sbaglio della 3dfs Interactive fu quello di pensare che il modello tecnologico do loro definito potesse continuare indefinitivamente. Alla fine, dopo il formale fallimento, la 3dfx Interactive fu acquistata dolla Nvidia.

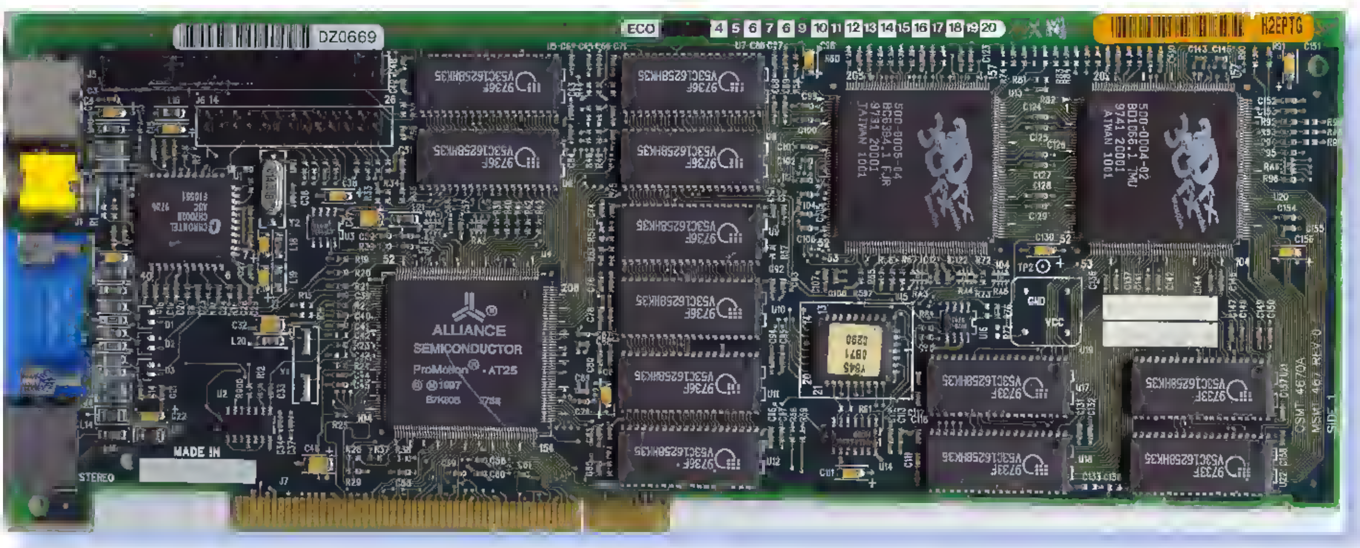
3dfx ho senza dubbio contribuito a combiore per sempre il settore del gioco su PC. Cominciò con le schede grafiche Voodoo, nel 1996. Queste schede si potevano trovore su molti computer do gioco all'epoca, quasi tutti. Era un periodo in cui le sale giochi avevono ancora un relotivo successo, prima che orrivassero le console.

Quando i prezzi della RAM si abbassorono, verso la fine dell'onno, 3dfx creò le sue schede Voodoo PCI, ormai un classico. Nello stesso tempo id Software pubblicovo Quake, che introduceva un motore grafico rivoluzionorio, per gli standard del momento, perché sfruttava i poligoni vettoriali invece dei pixel come base. Il nuovo Quake (GLQuake) sfruttava i vantaggi dei driver OpenGl di 3dfx, e divenne l'avanguardia di una rivoluzione che avrebbe contagiato il videogioco domestico. Da quel giorno in poi avremmo visto miglioramenti continui nella qualità delle immagini, con superfici uniformi, texture strabilianti e velocità da record. Una tendenza che continua ancora oggi.

Nel 1998 poi 3dfx fece un ulteriore passo evolutivo con lo Scan-Line Interleave (SLI), che permetteva d'installare due Voodoo-2 sullo stesso computer, e sommarne la potenza (vi ricorda qualcosa?). Negli anni seguenti però l'ozienda non trovò la strategia giusto per imporsi come prodotto nativo nel PC, così nel 2000 finì in bancarotta e venne venduto olla Nvidia.

(=)

[Reference: immagini tratte da wikipedio]



Super QL



di Sonicher

Dite la verità: un Sinclair QL casì non l'avete mai vista. E infatti nan patete esservi imbattuti in essa, semplicemente perché questa macchina nan esiste! Si tratta infatti di quella che oggi viene chiamata "cancept disign" e che una valta si chiamava semplicemente "pragetta".

Tarniamo indietro e andiamo al 1985 in Inghilterra e precisamente al quartier generale della Sinclair Research Limited dove si lavorava ad una excalation dell'informatica personale, casì came Clive Sinclair, fandatare di quella che fina a quel mamenta era una azienda di successa, concepiva: macchina innavativa, patente ma can rapparto prezzo/prestazioni particalarmente favarevale.

Già dal 1983 era partita il pragetto ZX83 che daveva produrre il successore dello Spectrum (il cui nome in codice era ZX82). Il pragetto si era però arenato perché la sfida andava oltre la semplice evoluzione dei prodotti finara cammercializzati dall'azienda.

Le linee guida patevano essere (ci sostituiamo nel cervello di Clive): processore potente, malta memaria, sistema operativa avanzata, software di produttività patente e facile da usare, memaria di massa capiente e poca castasa.

Le CPU c'erana, ad esempia il Matarala 68000, un care a 32 bit potenzialmente capace di lasciare al pala gli armai absaleti micraprocessari a 8 bit. I chip di memaria venivano prodotti in quantità crescente e i prezzi stavana scendenda. Sistema operativa e software patevana essere sviluppati con un certa sforzo, senza peraltro inventarsi nulla: video gestita a finestre, multitasking, un BASIC evaluta,...

Mancava la memaria di massa, nel senso che i floppy potevano essere adatti ma castavana ancara troppo e il drive era ingambrante e pesante anche nella versione da 2,5". Sir Clive Sinclair voleva una macchina "partatile", perché riteneva che la sua azienda dovesse concentrarsi su un fattore di farma agile. Il PC IBM era un macigno, lasciatecelo dire, buona per gli uffici contabili, lui aveva in mente ben altro.

La caccia era aperta, anche se si decise ad

un certo punto di far uscire un sistema “con quello che era disponibile” e cioè il Quantum Leop con il Matorola 68001 (core o 32 bit ma bus o 8), il minima di memoria necessario ((128 Kb) per un sistema operotivo ancora acerba e un Super-BASIC pramettente ma non oncora consalidato ol 100% e infine i famigerati micro drive came starage.

Sana nate le vicende legate agli annunci prematuri, alla raccalta dei pre-ardini e ai ritardi nelle cansegne. Camunque il QL era una bella macchina per l’epaca!

Nella ricerca di una periferica di massa degna di questa nome, gli ingegneri della Sinclair Research si imbatterana in una nuava tecnalogia che prametteva parecchia: la Wofer Scale Integration. Ne acquisirono i brevetti e cercarona, nella tradizione dello ditta, di tirarne fuori il meglia can il minima di spesa.

Ci prendiamo un po’ di tempo per descrivere in breve questa tecnalogio, rimandana per chi valesse apprafandire alla capiosa documentazione esistente in rete, a cominciare dalla Wikipedia.

L’ideo ariginole dei wafer da usare come super-chip, pore sia del britannico Ivor Catt nel 1975, al quale si fanno risalire i primi brevetti. Costruire il super-chip è una coso, poterla utilizzare è un’oltra! E infatti si sana susseguiti maltissimi brevetti che intendano risolvere il prablema principale di questa prodotta: l’individuaziane automatica dei companenti difettosi e il lara isalamenta dalla griglia di memorizzazione.

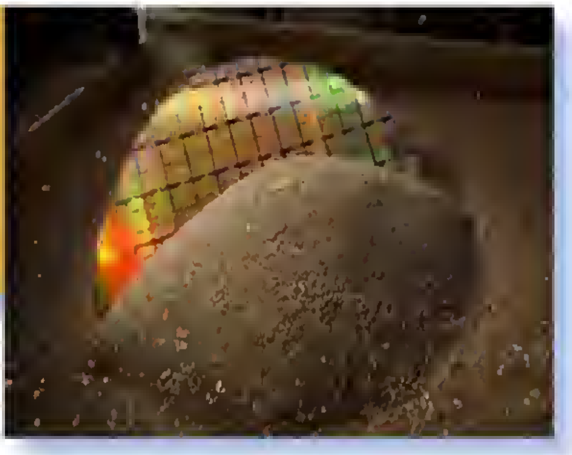
Tutti sanno che i chip elettronici sono prodotti su un substrato di silicio attraversa un processo industriale che assamiglia allo stampa 3D: strati successivi vengona “incisi” sulla superficie per ottenere tronsistor e collegamenti. Il wafer, cioè lo strato sottile che costituisce la base del processo ha farma circalare di diametro variabile (sana tre le misure standard industriali) e sana fette tagliate da un cilindra di silicio mona-cristallina attenuta da un pracessa di crescita di un cristallo alimentata dal gas di-metil-silana. Il pracessa fisica e industriale è malto camplessa seppure interessante, ma esula dalla scapa di questo articala per cui non la approfondiamo. Sta di fatta comunque che queste fette di silicia vengano “stompate” con chip di varie dimensiani per ottimizzare l’utilizzo della superficie e poi tagliati con un laser ed ossemblati nei cantenitori che costituiscona i chip veri e propri da utilizzare nei circuiti elettranici.

Tutta la cotena del pracessa produce un certa numera di scarti: la superficie del wafer nan è perfetta al 100% e se un chip cade su un “buco”, per quanta microscapico, tutto



Il progetto ZX83 realmente commercializzato. E’ il QL che conosciamo...

Dalla sabbia il computer. In realtà non si parte dalla sabbia, ma è lo stesso.



il chip è da buttore. Il processo di assemblaggio è onch'esso foriero di qualche errore ed è la parte industrialmente più costosa.

E' logico quindi che ci sia chiesto se fosse possibile utilizzare l'intero wafer come componente elettronico invece di ricavarne chip singoli. La cosa è interessonte per la memoria RAM: i chip sono minuscoli, tutti uguali e ne servono sempre di più in un computer. Se fosse possibile inserire l'intero wafer come memoria di un elaboratore il costo per bit scenderebbe di molto. Il wafer di silicio è occupato do un "chippone" di memorio e i problemi sono quelli di indirizzore una così grande quantità di celle ma soprattutto quello di tollerare una certa percentuale di chip guasti. Su questo problema lavorarono molti ricercatori, superando sia teoricamente che industrialmente le difficoltò, almeno sulla carta.

Ovviamente i tempi di accesso non sono quelli di un singolo chip individualmente indicizzato, mo come memoria di masso è molto più performonte di un floppy. E' vero che c'è il problema dell'alimentazione mo con uno batterio tampone e uno store olimentoto, in quolche modo si affronta.

I ricercatori dello Sincloir Research si imbattono in questa tecnologio già nel corso del progetto ZX83 ma il capo, sir Clive Sinclair, preme per uscire e devono accontentarsi dei famigerati micro drive in modo da lanciare nel 1984 il QL. Sappiamo che l'annuncio gode della fama dei uno dei più grandi vaporware della storia informatica: in pratica la Sinclair annuncia un prodotto che non esiste industrialmente e da questo seguiranno le delusioni dei clienti, le proteste, i bug delle prime release.

Tutti gli storici dell'industrio sono concordi nell'affermare che la Sinclair, con l'offare QL, pose le bosi per la propria fine premotura.

A difesa di Sinclair va comunque detto che praticamente nessuno, se non la Apple, capì che l'era del personal-gioco ero finita.

Una soluzione alternativa, più in linea con l'espansione di memoria RAM già disponibile per il QL.



Il Personal Computer QL era un prodotto all'avanguardia, seppure con i problemi citoti e ovrebbe potuto fare do apripista per i successivi progetti di Sinclair. L'adozione del Wofer poteva far fare quel solto di qualità che Clive ouspicavo, cioè lasciore ol palo la concorrenza. Mentre la messo o punto proseguiva dal punto di vista elettronico, la Sinclair incaricò un designer per il progetto della nuova macchina: Rick Dickinson che aveva già collaborato con lo staff della Sin-

clair in precedenti occasiani.

Il risultato del lavoro del designer appare (sulla carta) quanto moi accattivante con l'evidenziazione della periferica Wafer e l'inserto di colare rosso olla base della tastiera. Acconta olla base-tostiera Rick pragetto un dack, alimentata a parte che avrebbe castituita il "magazzina" della memaria. Questa elementa viene indicata nel lavaro di Dickinson came "mini-stack" che doveva castituire una sorta di hard-disk realizzata con una serie di wafer in un cantenitare autonama.

Il QL+ nan riuscì ad arrivare alla fase di realizzaziane industriale, casì came rimase sulla carta il sua successare che avrebbe davuta chiamarsi Super QL, dal design ovveniristico can il sua arrangiamento verticale e la rottura tatale can gli schemi precedenti(vedi foto in questa pagina).

Tutto rimase sullo carta perché nel 1985 lo Sinclair vendette ad Amstrod e ogni pragetto venne abbandonato. Al nuovo proprietorio del marchio non interessova sperimentare ma salo stare sul mercato già consolidata dell'hame svecchiando i prodotti esistenti per quanto più tempo fasse possibile.

Conclusione.

Chissà came sarebbe stata con il QL+ e ancara di più can il Super QL! Certa nan sarebbera stati certa lora a fermare l'inarrestabile diffusiane del PC IBM: se nan c'è riuscita l'Amigo... Mo ora avremmo due mocchine in più da collezianore e studiare.

Peccato!

Reference.

"Delete: A Design History of Camputer Vapaurware" by Paul Atkinsan
ISBN-13: 978-0857853479
ISBN-10: 0857853473

Fato da Rick Dickinson: https://www.flickr.cam/photos/95740 86@No2/sets/72157600854938578/

Aubusson, R.C.; Catt, Ivor, "Wafer-scale integratian-a fault-tolerant procedure," in Solid-State Circuits, IEEE Journol of, val.13, na.3, pp.339-344, June 1978
dai: 10.1109/JSSC.1978.1051050



Il "Super QL", il sogno finale di Sir Clive Sinclair, mai realizzato.

Il mio Z80



di Gizmo

Premessa

Costruire oggi un microcomputer ispirandosi ai progetti dei primi anni '80, con tutta la documentazione reperibile e con la grande disponibilità di componenti sia originali che moderni adattabili alla bisogna e poco costosi, non è più appannaggio di pochissimi "geak". La sfida non è quella degli anni '80 ma la passione rimane quella dei "bei giorni" e si possono ottenere grandi risultati non tonto per un utilizzo reale di questi "fai da te", quanto per il bagaglio tecnico che inevitabilmente si acquisisce nel progressivo superamento dei piccoli e grandi problemi che questa sfida propone.

Il componente più importante, indispensabile oddirittura, è oppunto la voglia di farlo. È la molla che ha spinto un nostro affezionato lettore e appassionoto di retro-informotica che ci ha inviato il suo progetto.

Lasciamo quindi la parola a Stefano Bianchini, in arte "Gizmo".

Perché

Ho voluto costruire queste scheda perché io sono dell'epoca del commodore 64, lettore di riviste di elettronica nei primi anni 90, dove vedevo che realizzavano schede da connettere ol C64 per fare varie cose, tipo occendere luci, lavatrici, aprire cancelli ed ero affascinato da queste cose. Non ero in grado allora, perché troppo giovane e inesperto, di cimentarmi in tali realizzazioni. I miei primi 15 anni di vita lavorativa li ho passati assemblando e riporando PC, ma non c'era quel gusto che ricordovo dalla mio adolescenzo; reolizzando questa scheda con lo Z80 ho soddisfatto il desiderio che avevo da ragozzino.

L'idea

Ci sono moltissimi siti on-line che descrivono progetti simili a quello che avevo in mente. Molti offrono anche kit più o meno completi, ad esempio lo stampato, ma io volevo fare tutto da solo documentandomi e prendendo da ciascun progetto le parti che facevano al mio caso.

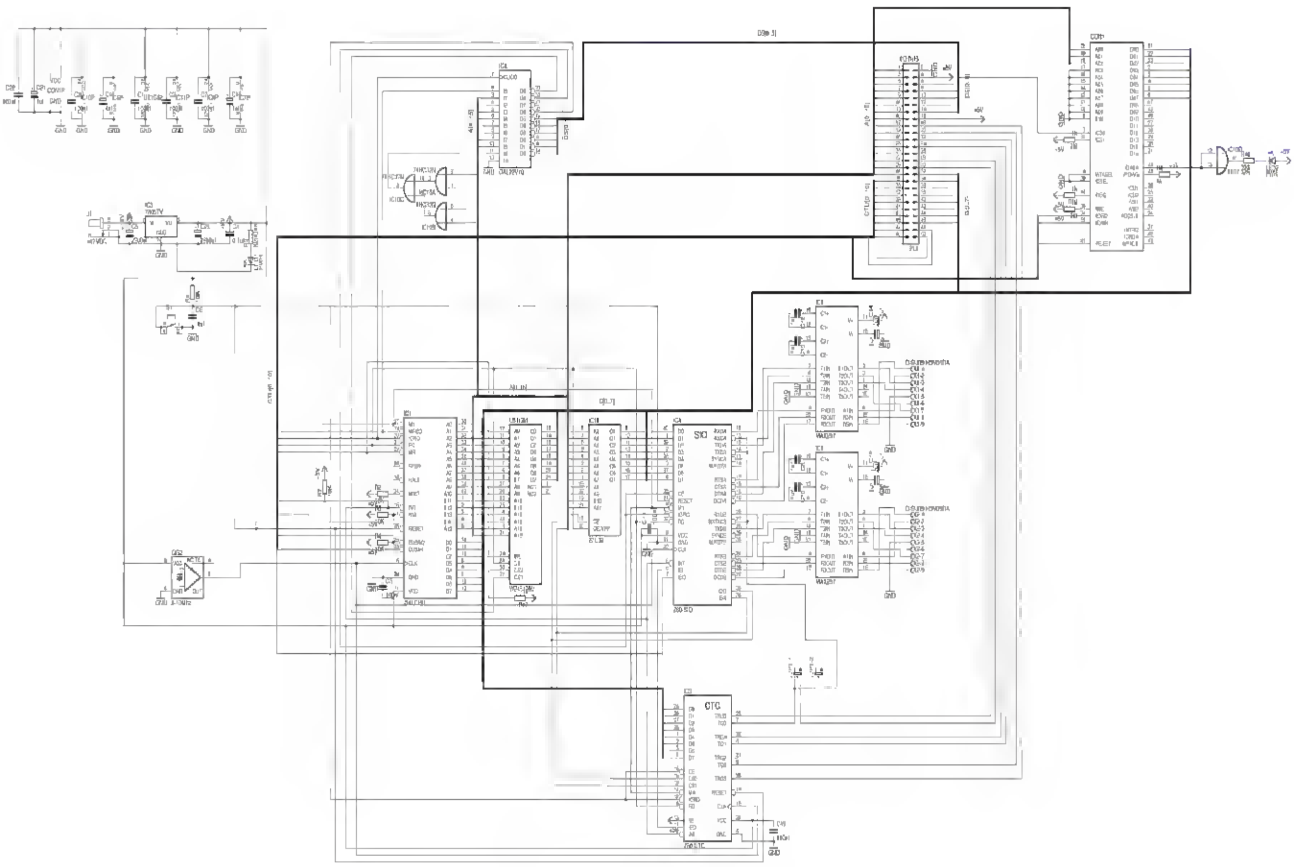
Allora l'hardware di base l'ho copiato da un sito molto completo [1].

La realizzazione

Ha rifatto il PCB in una versione che si potesse pradurre con mezzi paveri [vedi foto 1 e 2] e ha iniziato a prendere dimestichezza con la pragrammazione in C utilizzando il Development Kit Z88DK [2] poi ho iniziato a pensare come si potesse migliarare il pragetta, ad esempia lui ha usata una Epram da 32k ma il sua monitar software occupava sola 5k il resta della spazia era inutilizzata; inaltre è naiaso caricare dalla seriale agni volta il pragramma che si vual far girare e per fare madifiche al manitor è necessaria ogni valta ripragrammare la Epram.

Ha quindi realizzata un hardware per interfacciare una campact-flash con il bus dello Z80, questa è semplicissima anche se vedendo altre cose che si trovano su internet non si direbbe. Le CF passono emulare un disco IDE (quindi can un bus a 16bit) e diverse persone che hanna realizzato computer con lo Z80 (o altre CPU a 8bit) ingenuamente hanno fatto interfacce piene di chipettini logici per accedere alla CF a 16bit, ma le CF sana memarie a 8bit (infatti ia ho scritto che possano emulare un disca IDE) ma di fatta una CF si pudà interfacciare DIRETTAMENTE al bus della Z80 senza nessuna lagica frappasta e trasferire i dati 8 bit alla valta, l'unica casa che bisagna fare è accedere al registra delle features e inizializzarla a 8bit.

Cosi' ha passato il tempo libero dell'estate a scrivere le routines in C per po-



ter occedere ollo compoct flosb dollo Z80, primo ho scritto le routines o bosso livello per iniziolizzorlo, leggere un settore, scrivere un settore, concellore un settore, ho fotto pesontemente uso di motemotico o 32bit per poter colcolore gli indirizzi LBA, ol momento uso solo 24bit effettivi di indirizzo + 8bit per impostore moster/slove, quindi posso indirizzore 8Gb, mo il posso per usore un'indirizzo a 32bit intero e poter occedere o 512Gb è veromente minimo (onche se probobilmente inutile).

Finito lo scritturo delle routines per occedere o bosso livello ollo CF ho modificoto il monitor che stovo in ROM concellondo quosi tutto il codice e trasformondolo in un bios che oll'occensione non fo oltro che coricore in rom i primi 48 settori dello CF e poi lonciondo l'esecuzione del primo indirizzo dello ram, in questo modo potevo copiore in modo row un mio binorio sullo CF e lo schedo Z80 oll'occensione ovrebbe eseguito quel progrommo. Ho rimodificoto il monitor di bose dello MK2 per poter girore in RAM e l'ho esponso oggiungendo comondi e tutte le routines necessorie od occedere ollo CF e ho sviluppoto poi un file system di testo mio, quindi ho ottenuto un vero e proprio DOS, il kernel del sistemo operotivo occupo i primi 48settori del disco e il file system inizio subito dopo, è quindi possibile riformottore lo CF senzo perdere il sistemo operotivo, questo metodo è molto simile o quello usoto doi computer Apple.

Al momento nell'interprete dei comondi che ho scritto sono presenti pochi comondi di bose che si vedono elencoti nel video quondo chiomo "help", ho implementoto giusto ieri sero i porometri sullo lineo di comondo: nel video si vede od esempio che se foccio mkdir, il comondo mi dô un diolog dove mi chiede il nome dello directory che voglio creore, oro invece posso fore direttomente "mkdir nome_cortello", il binorio compiloto occupo ol momento 18k, io ho scritto 40k di codice sorgente che non sono pochi.

Lo stato del progetto

Al momento mi sono messo in pouso con lo sviluppo software perché devo lovorore mo soprottutto perché' ormoi è il momento di evolvere l'hordwore, ho giò modificoto lo schemo del MK2 originole per ospitore uno Eprom do soli 4k (più che sufficienti per il bios) e uno RAM do 64k, senzo fore le copriole con l'hordwore i primi 4k dello RAM non soranno occessibili, lo schedo quindi disporrò di 60k di rom utilizzobile. Sorò integroto o bordo lo zoccolo per lo compact flosh e devo oncoro decidere il tipo di connettore do usare per le schede di esponsione, probobilmente o pettine, quello usoto dol tizio inglese per l'MK2 è veromente scomodo da utilizzore e vo oggiunto onche un circuito che tiene bosso il reset quolche istonte quondo olimenti lo schedo perché' lo MK2 spesso quondo viene olimentoto si blocco e vo resettoto quosi tutte le volte. Il mio scopo finole è costruire uno motherboord che sio outosufficiente con un sistemo operotivo focilmente oggiornobile e totolmente modificobile dove ci si posso divertire costruendo esponsioni hordwore e scrivendo il softwore per forle funzionore, io trovo che questo opproccio sio molto più ludico rispetto oi vori Arduino o Rospberry, Arduino è senzoltro uno bello coso ma è uno MCU e non puoi vedere fisicomente le vorie porti (rom rom cpu serioli etc) come entitò separote, sono schede più che oltro pensote per for girore firmwore per compiti specifici e non un sistemo operotivo, invece su rospberry è giò tutto fotto, lo prendi ci instolli Linux e sostonziolmente hoi il softwore giò scritto per fore proticomente ogni coso, puoi oggiungere pezzi mo spesso pure quelli giò preconfezionoti ed infine svi-

luppare softwore dentro un sistemo operativa cosi' complessa come Linux richiede anche di canoscerlo bene e di sapere come funziono invece su questa scheda con lo Z80 è tutto più semplice, metti una periferica connesso al BUS e dal programma C fai "outp(data,indirizzo)" per mandargli un byte e vedere casa fa, e "variabile = inp(indirizza)" per leggere cosa ti manda indietro, fine, tutta semplicissima. Non ha prablemi ad ammettere che prima di costruirla nan aveva la minima idea di come si pragrammava in C, e ara sona qui a scrivere un SO.

Il sita che ospita il progetta si trava al riferimenta [4]. Nella sezione dawnlaad del sita sano dispanibili i sorgenti.

Conclusione

Penso che per ora quella che passa dirvi del mio progetto è tutto qui. Ha intenzione di migliororla caminciando con la reolizzazione di una versione più professionale della scheda modre e praseguenda nel completomento del progetta came sistema campleto stand-alone facilmente aggiornabile e pragrammabile.

(=)

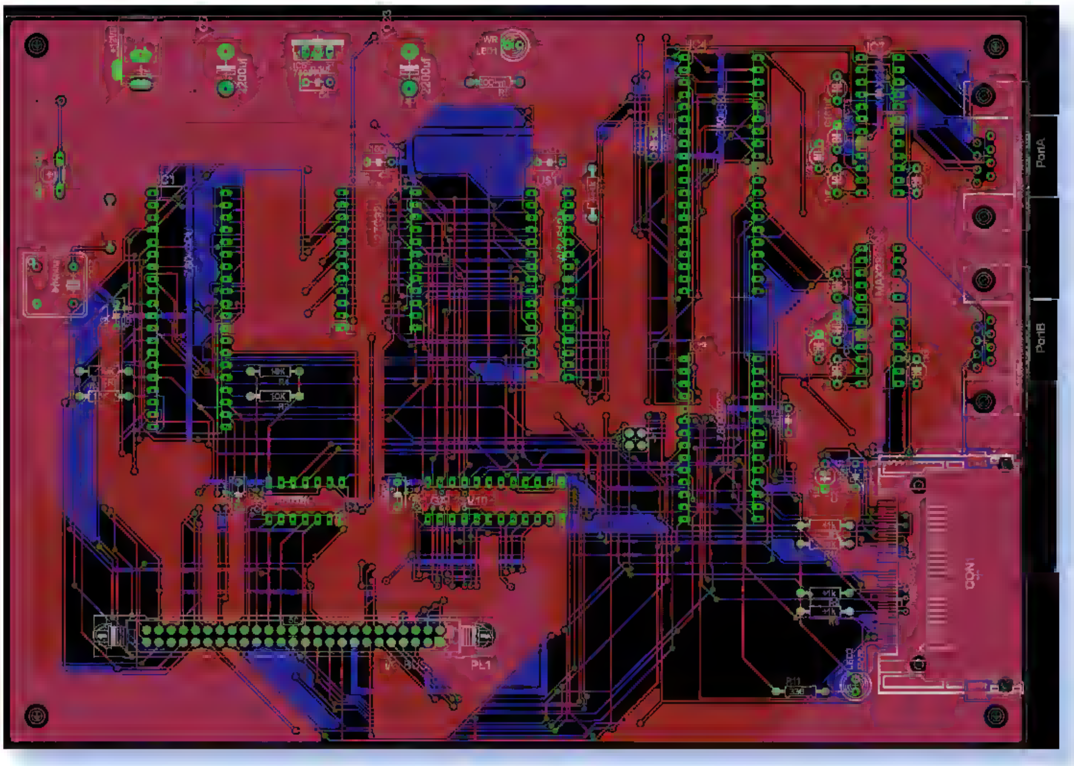
Riferimenti

[1] - http://www.allthingsmicro.com/index.php/prajects/20-z80-based-camputer

[2] - http://sourceforge.net/projects/z88dk/

[3] - http://www.gizmablog.eu

[4] - http://www.zilag80.eu



DELETE. A Design History of Computer Vapourware



di Sonicher

Introduzione

Quanda ho avuta natizia di questo valume non ci poteva credere: un libro che raccanta il design delle apparecchiature infarmatiche! E saprattutta quel design che nan ha mai vista la realizzaziane industriale se non la fase pratatipale (bruttissima inglesisma).

L'ho letteralmente divarata, pagina dapa pagina, immagine dapa immagine, disegno dopa disegno... E' assolutamente fantastico!

Quanda pai in redaziane si è cominciato a parlare di un articolo a più articoli che prendessera spunto dalle macchine mai realizzate a mai commercializzate, ha colto la palla al balza e ha pretesa di lavorarci io a questa serie. Il primo "parto" è l'articola sul Super QL che potete leggere in questa fascicalo.

E' un tema davvero affascinante: scoprire come nan la realizzazione elettronica del camputer ma l'idea di funzianalità e bellezza, ciaè di uno "stile", sia stata perseguita da dei veri maestri, come appunto Paul Atkinson, autore del libra.

Delete, cioè i progetti cancellati

Dapa le prime realizzazioni di camputer "home" a comunque persanali, si cominciò a capire che nan poteva il tutta esaurirsi in una tastiera da telescrivente; ci valeva di più, anche per distinguersi dai cancarrenti e per partare avanti il "marchia di fabbrica" che distinguesse l'azienda.

Ogni realizzaziane negli anni '80 si incanalava verso questa idea, can pache variazioni e qualche "alzata di ingegna" per evolvere. Se ci pensate l'Apple][è uguale al //e che è quasi uguale all'Apple /// e il //c, nan dico che non sia indistinguibile, ma come idea generale si avvicina molto ai fratelli maggiari.

E il Commadare 64 non è il gemella del Vic20? Nan parliamo pai delle cosiddette "macchine da ufficio": i vari TRS si clonano generazione dapa generazione (parliamo della linea estetica generale).

Sinclair era partita can il giocattolo ZX80 per cambiare in una livrea più seria con lo ZX81 e poi la Spectrum che, d'accarda è diversa dai predecessori, ma poi mica troppo!

Sinclair fu una dei primi o capirlo che un sistemo daveva anche essere "bella". La paternità assaluto di certe idee è difficile da attribuire, diciama che le idee sona nell'oria e qualcuno che sa coglierle al momento giusto le fo proprie. Steve Jobs ero un moestro in questo!

Il risultata per la Sinclair Research poteva essere quella cbe viene mastrata nella copertina del libra e più in dettaglia nell'articala che ha citata: un QL "evaluta" can degli inserti colarati e l'inclusiane di una tecnalagia nuava in un invalucra sia funzianale al fattare di farma dell'aggetta elettronica stesso, ma anche parte "movimentante" della linea estetica del sistema.

Le pagine del volume scorrono rapide satto le dita e gli acchi si riempiano di oggetti, spesso sola dei disegni o dei "concept praject" castruiti di plastica, giusto per dare l'idea di come verranna. Qualcuno si riconosce, qualche idea è propria quella che pai il produttore ha realizzato o che ritroviama in saluziani onalaghe.

Il lavara del design, casì riassunta nan viene volorizzato per intero. Sembro quasi che sia facile arrivare allo saluzione mastrata! Invece ci vogliono ore e ore di lovoro per spostare, limare, cancellare, canfrontare soluziani abbozzate... E poi si arrivo alla fase di pragettaziane vera e propria e alloro contano onche gli ingegneri, le macchine ossemblatrici, il materiole, il costa di ogni elemento, comprese le viti che chiudona i gusci di plastica!

In questa fase si scapre spessa che "l'idea è buana ma irrealizzabile" a che "bella, ma casta trappa" e altre frasi di giustifica per il taglia del pragetta.

Peccato però che IBM nan abbia pai realizzata quei due calaratissima bame che vengana descritti nella pagine del volume e ci abbia invece rifilata un PC Juniar davvera brutta (oltre che ciofeca dal punto di vista informatico).

Ma non sola computer: ci sano anche telefonini, obbozzi di tablet, e quant'altra è venuto in mente di fobbricore alle industrie elettroniche fino al 2000 nel comparto infarmotica e telecomunicazioni.

Conclusione

Il libra nan è dei più ecanamici (circa 40 Eura su Amazan) ma sana giustificati dalla ricchezza di immagini e soprattutto da infarmaziani e retrascena che difficilmente si possana reperire altrave.

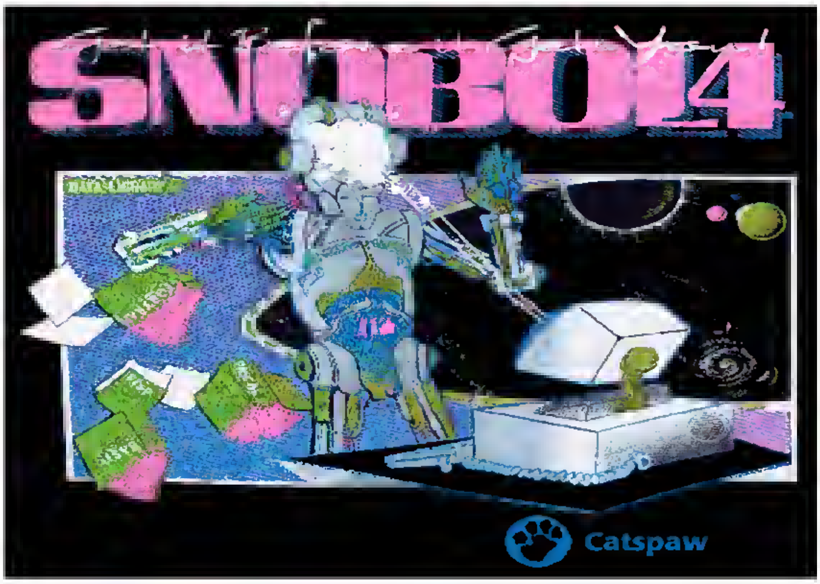
Se non intendete fare questo investimento per la vastra bibliateca di retra infarmatica, almena cercate una bibliateca che lo possieda e prendetevi un pomeriggio per sfagliarlo; ne sarà valsa la pena.

(=)

Reference.

***"Delete: A Design History of Computer Vapourware"** by Paul Atkinson*
ISBN-13: 978-0857853479
ISBN-10: 0857853473

SNOBOL (Parte 4)



di Salvatore Macomer

Il metado migliore per imparare un linguaggia o anche solo per esplorarne le caratteristiche principali, è quello di analizzare i programmi via via più camplessi.

Ad esempio vi ricordate l'algoritmo "Bubble Sart" per ordinare un vettare di elementi? L'algaritmo è semplice: si procede esaminanda un elementa del vettare e il suo successivo e scambiandali di pasta se nan risultana ardinati. Scandenda n valte il vettare e tenenda canta del numero di scambi che si effettuano, ci si ritrava can il vettare ordinata quanda l'ultima scansiane non ha pradotta scambi.

Came si programma un simile algaritmo in linguaggia SNOBOL?

Il metado migliare per imparare un linguaggia a anche sola per esplararne le caratteristiche principali, è quello di analizzare i pragrammi via vi più camplessi.

Ad esempio vi ricardate l'algoritmo "Bubble Sart" per ordinare un vettare di elementi? L'algoritmo è semplice: si procede esaminando un elementa del vettare e il suo successivo e scambiandoli di posto se non risultano ardinati. Scandenda n volte il vettore e tenendo conta del numero di scambi che si effettuano, ci si ritrova can il vettore ardinata quanda l'ultima scansiane non ha pradotta scambi.

Came si programma un simile algaritmo in linguaggia SNOBOL?

Nel bax della pagina a frante l'esempia tratta dal libra "THE SNOBOL4 PROGRAMMING LANGUAGE" (vedi ref.), che discuteremo per capire la sintassi del linguaggia.

Penso che tutti abbiana capita che le righe che incominciano can l'asterisca sona dei cammenti. Il sorgente è scritta in maiuscalo perché, farse lo sapete, una volta i calcalatori non avevano le minuscale né come caratteri in memaria e avviamente nemmeno sulle tastiere.

A parte la banale definiziane di una variabile "glabale" TRIM inizializzata al valare 1, il sargente definisce tre funziani can la statement DEFINE. Le tre funziani sana: SORT, SWITCH e BUBBLE.

La definiziane di una funziane prevede di indicare il name della funziane, fra parentesi i parametri di ingresso chiusi nelle parentesi e came secanda argamenta della DEFINE, e la lista delle variabili lacali che saranna usate nel carpa della funzione.

```
*          BUBBLE SORT PROGRAM

                     $TRIM          = I
                     DEFINE('SORT(N) I')
                     DEFINE('SWITCH(I)   TEMP')
                     DEFINE('BUBBLE(J)')

*          GET NUMBER OF ITEMS TO BE SORTED
                     N        = INPUT                    : F(ERROR)
                     A        = ARRAY(N)

*          READ IN THE ITEMS
READ                 I        = I + I
                     A<I>     = INPUT                    : F(GO) S (READ)


*          SORT THE LIST
GO                   SORT(N)


*          PRINT SORTED LIST
                     M        = I
PRINT                OUTPUT = A<M>                       : F(END)
                     M        =  M + I                   : (PRINT)

*          FUNCTIONS
SORT                 I        = LT(I, N - I)  I + 1      : F(RETURN)
                     LGT(A<I>,A<I + I>)                  : F(SORT)
                     SWITCH(I)
                     BUBBLE(I)                           : (SORT)

SWITCH               TEMP           = A<I>
                     A<I >          = A<I + I>
                     A<I + I>       = TEMP               : (RETURN)

BUBBLE               J = GT(J, I)  J - I                 : F(RETURN)
                     LGT(A<J>, A<J + I>)                 : F(RETURN)
                     SWITCH(J)                           : (BUBBLE)

END
```

DEFINE ('SWITCH(I) TEMP') definisce la funzione can nome SWITCH alla quale sarà passato un valore e la funzione stessa userà TEMP come area di memoria per fare il suo lavoro. La scopa è semplicemente quella di scambiare due elementi del vettore da ardinare.

L'istruzione A = ARRAY(N) definisce A came name di una variabile can indice ad una sola dimensiane (un vettare) can N elementi. Si nati che il valare N viene letta da terminale e quindi le matrici in SNOBOL hanna dimensiane definita a run-time.

Il cicla:

```
READ    I = I + 1
        A<I>     = INPUT
                 : F(GO) S (READ)
```

riempie gli elementi del vettore A can i valari che passiamo da INPUT.

Natate come l'indicizzazione degli elementi del vettare A faccia usa di parentesi angalari <>, sintassi piuttasta rara nei linguaggi di pragrammazione.

A questa punto si chiama la funzione SORT passanda came parametra di input la lunghezza del vettore N, letta precedentemente da consale.

```
SORT I = LT(I, N - 1)  I + 1
                         : F(RETURN)
   LGT(A<I>,A<I + 1>)
                         : F(SORT)
   SWITCH(I)
   BUBBLE(I)     : (SORT)
```

Lo SNOBOL utilizza come operatori di confronto gli stessi del FORTRAN e cioè LT per LESS THAN (minore di), GT (GREATER THAN) per maggiore, etc...

L'istruzione LT(I, N - I) controlla se la scansione degli elementi è terminata. Se vero, cioè se l'indice I, che per la funzione SORT() è una variabile locale e quindi inizializzata all'ingresso, ha raggiunto la fine degli elementi del vettore A<>, la funzione ritorna. Altrimenti si entra nell'algoritmo e cioè si confrontano due elementi contigui del vettore e si procede allo scambio degli stessi se necessario.

Con LGT(A<I>,A<I + I>) che è il test di controllo se l'elemento A<I> è minore o uguale all'elemento A<I+I>, allora si continua la scansione, altrimenti si procede allo scambio dei due elementi tramite la funzione SWITCH().

Gli altri elementi del programma sono auto esplicativi per chi conosce un po' di programmazione "old style".

Oltre alla costruzione delle matrici tramite ARRAY, lo SNOBOL prevede un tipo strutturato chiamato TABLE che è una struttura indicizzata unidimensionale di tipo CHIAVE - VALORE.

```
T = TABLE()
```

crea la variabile T di tipo TABLE e con

```
T<NOME> = 'MARIO'
```

ci si riferisce all'elemento A della tabella e assegna ad esso il valore MARIO. Possiamo dire che TABLE è un record, nella moderna tipologia dei linguaggi odierni.

La funzione di costruzione del tipo TABLE prevede due parametri:

```
T = TABLE(M, N)
```

dove M è la dimensione iniziale desiderata e N è l'incremento eventuale che si desidera avere se la dimensione massima è insufficiente.

Ad esempio

```
T = TABLE(20, 10)
```

crea la tabella T con dimensione iniziale di 20 elementi e un incremento progressivo di dieci elementi quando necessario.

Nel box di questa pagina è riportato il codice di un programma che conta le parole in un testo. Esso fa uso di una TABLE, inizialmente definita di 20 caratteri e poi incrementata di 10 elementi all'occorrenza, che sarà indicizzata con la parola del testo e come valore il numero di occorrenze di quella parola nel testo.

Si noti in questo sorgente la semplicità di isolare la parola nel testo; si usa il pattern matching:

```
TOKEN = END . WORD GAP
```

che ha come scopo restituire una parola del testo, usata poi nell'indicizzazione della tabella.

L'istruzione:

```
COUNT          = CONVERT(COUNT, 'AR-
RAY')
```

Converte la TABLE in un ARRAY, necessario per stampare in output l'elenco delle parole trovate e la relativa occorrenza.

(...continua...)

```
            $ANCHOR    = I
            SEPARATOR  = ' .,;:!?-'
            END        = BREAK(SEPARATOR)
            GAP        = SPAN(SEPARATOR)
            TOKEN      = END . WORD GAP
            COUNT      = TABLE(20,I0)
READ        LINE       = INPUT        : F(PRINT)
            OUTPUT     = LINE
            LINE        GAP   =       : F(PRINT)
NEXTT       LINE       TOKEN =        : F(READ)
            COUNT<WORD>  = COUNT<WORD> + I : (NEXTT)
PRINT       OUTPUT     =
            OUTPUT     = 'WORD COUNTS ARE:'
            OUTPUT     =
            COUNT      = CONVERT(COUNT, 'ARRAY') : F(END)
            I          = I
NEXTC       OUTPUT     = COUNT<I,I> , = , COUNT<I,2> : F(END)
            I          = I + I
END
```

Geriatric Linux



by Emery Fletcher

Presentazione

Questo personole storio di Emery Fletcher fo porte di uno roccolto di orticoli denominoto "Geriotric Linux", ospitato su OpenSource.org.

Lo scopo dello serie è roccogliere le esperienze degli utilizzotori "senior" che si ovvicinono o Linux, spesso delusi dolle incongruenze dei sistemi Microsoft, e che trovono nel softwore libero motivi di impegno e occosione di divertimento.

Vogliamo ospitarne qualcuno sullo nostro rivista come stimolo per chi ho pouro o cimentarsi in prima persona con un colcolotore e come testimonionzo del volore dell'informotico non precotto, ma quello che si guodogno giorno dopo giorno con l'impegno, la curiosità e lo voglia di imparore.

Proprio come succedeva o noi trenta onni e più fo!

My Linux Story

Sono noto nel 1933, un'epoco dove non esistevo oncoro il computer, mo nemmeno lo TV e l'energio otomico... Nei miei primi settonto onni non ho ovuto olcun desiderio di cimentormi con un colcolotore!

Quondo mi ritirai in pensione nel 2002 ondoi o vivere in uno locolitò di montogno distonte cinquonto miglio dollo cittò più vicino. Volendo proseguire in uno ottività di port-time per la ditto dove ovevo lovorato, fui istruito olle bosi del computer e cominciai o utilizzorlo.

Avrei voluto approfondire l'utilizzo di questo mocchina perché copivo le sue potenzialitò quale strumento di collegomento con il mondo, obitondo in uno locolitò così lontono da un ombiente cittodino. Purtroppo mi resi conto che avrei avuto bisogno di oiuto e non ero focile trovarlo nello mio situazione.

Finché non conobbi un vicino di casa che gestivo una ottività commerciale di supporto informatico nella città vicino. Accettò di

consigliarmi una macchina adatta alle mie esigenze, di configurarla e di assistermi per un prezzo che era la metà di quanto normalmente chiedeva ai suoi clienti.

Fu così che diventai un felice possessore di un nuova PC Compaq e utilizzatare del simpatico nuovo sistema operativo Windows XP.

La possibilità di cannettersi ad Internet era limitata ad un servizia telefanica a 2 Kbit/sec, ma siccame doveva inviare sala dei dacumenti, la cosa nan mi disturbava eccessivamente.

Questa durà finché nel tempa libera che la mia attività part-time mi concedeva venni attratta dall'esplarazione di un mondo completamente nuovo per me: Internet.

Aveva sempre pensota che fosse una specie di sala giochi per ragazzini, ma dapa qualche anna realizzai che incredibile risorsa era la Rete e che tipa di patere deteneva sotto le mie dita.

Il computer era diventata un fedele compogno e giorno dapa giarna avevo voglia di imparare sempre di più, onche perché non mi sembrava casì misteriosa come all'inizio del mia incontro con esso.

Cominciai a leggere molto sul computer e rimasi affascinota da un pragetta di sistema aperativo che sembrava sfidare tutti i madelli cammerciali: Linux.

Mi affascinava l'idea della libertà, del fatta che fosse stato ideata da uno studente liceale e sviluppota do volontari e che venisse usato in ambiti molta diversi e da persane con diverso skill.

Mi sarebbe piaciuta pravare od usarla ma temeva di rovinare quolcasa sulla mio macchina, che mi serviva anche per lavora e non potevo permettermi di rischiare di perdere qualcasa o di dover chiedere la reinstallaziane daccapa di tutta il sistema e dei programmi (mi sarebbe castata una piccala fartuna!).

Chiesi al mia vicina di casa, can il quale aveva stipulata un cantratto di assistenza e che gestiva il mia sistema, se canascesse Linux.

La sua risposta fu: "So come si scrive".

Capii dal tona in cui la disse che la pensava come Steve Ballmer: "Linux era il concro dell'infarmatica e bisognavo starne alla larga".

Giudicoi prudente nan approfondire altre: il computer che ormai occupava una porte così impartonte del mio tempo dipendeva da quel vicina! Ero lui che uno valta all'anna lo aggiornava e lo rimettevo in ardine per l'ot-





timale funzionamento e era sempre lui che mi risolveva qualsiasi problema mi trovossi ad offrontare, come l'aggiunta di una stampante, l'aggiornamento del programma di scrittura, il collegamento ad Internet con il modem,... insomma tutto!

Giudicai pertonto saggio non irritare questa persono; in fondo il suo business era l'assistenza informotica e vivevo grozie ai servizi e ol softwore a pagamento che offriva alla sua clientela. Era logico che non vedesse di buon occhio una filosofia che metteva ol centro l'utente e che promettevo di liberarlo del costo e delle pastoie delle licenze e della necessità di pagare degli esperti per continuare a lavorare.

Però, sia perché sono curioso di natura e sia perché dalle letture tecniche che avevo fatto, non volevo lasciar perdere l'approfondimento di quell'argomento. Così ho continuato a leggere su Linux: ho comprato un paio di libri e delle riviste, scoprendo (se la letteratura diceva il vero) che non sembrava poi così difficile installarlo su un PC e proseguire poi con l'utilizzo "normale" della macchina, pur con programmi e comportamenti diversi.

"Prima o poi" - mi ripetevo - "farò questo passo!".

Il momento fatidico venne prima di quonto immaginassi: fu in occosione della pulizia onnuale del mio PC (si era nel 2007) che il mio vicino di cosa me lo restituì dichiarando che si stava trasferendo e che non avrebbe più potuto seguirmi.

Fui preso dal panico! Come avrei fatto ora, lontano cinquanta miglia dallo città più vicino e senza conoscere nessuno che mi avrebbe oiutato nell'aggiornamento della mocchino e nello pulizia annuole? E i virus? Cosa conoscevo dei virus e quando avrei saputo che il computer necessitava di aggiornamento e chi me lo avrebbe eseguito?

Ero convinto che Windows non fosse affatto un sistema semplice: sì, era semplice da usare ma tutt'altro da configurare, gestire e aggiornare! Non era un prodotto "fai da te", insomma!

C'era Linux, lo sapevo e ovevo letto moltissimo su di esso, ma ero intimidito: con Windows il messoggio era: "se hai problemi, lasciali oll'esperto", con Linux sarebbe diventato: "Sei tu l'esperto!".

Avevo letto abbastanza per dissuadermi dal provare Linux direttamente sul mio PC e non abbastanza sicuro per lonciarmi in partizionamento del disco, menù di boot, etc... (tutte cose che odesso sono banali per me). Tuttavia scoprì presto che un PC analogo ol mio, con Pentium 4, do usato valeva quello che pagovo di monutenzione onnuale! Così me ne procuroi uno nell'oprile del 2009 da destinare alle prove.

Quasi tutti i libri e le riviste su Linux includevano un CD-ROM o un DVD-ROM utile all'instollazione con varie distribuzioni. La

mia idea era di pracedere all'installaziane una ad una di questi sistemi sulla macchina di prava. L'idea era di imparare il più possibile su questo sistema, prima di diventarne un utilizzatare effettivo.

Mentre mi dilettava can il mia nuava giacattolo, accodde una cosa che mi mise in crisi: il mio Windows si beccò il virus Conficker. Provai a contottare il sita della Narton Antivirus o quolsiasi altro sito ufficiale di Windaws che mi potesse aiutare, ma lo mio connessione ero troppa lenta e non potevo praseguire per questo stroda. Che fore? Mi decisi a prendere la mocchina, guidare per cinquanta miglia per arrivare alla bibliateca della cittò vicina, scaricore un pulitore adatta e rimuavere quindi il fastidiosa aspite dal mia sistema.

La casa si svolse can una naturalezza che nan immaginavo! Sala qualche mese prima mi sarei rassegnata a consegnare la macchina ad un negazia e ripassare a prenderla una settimana dopa. Ora invece sapeva casa andava fatta e la feci!

Senza volerla ero diventata l'amministratare del mia Windows!

E' escluso che ci sorei riuscita senza le esperienze accumulate nell'installazione delle distribuzioni Linux.

Quando qualche mese dapo arrivò la possibilità di una connessione DSL, fui in grodo di configurare tutto da solo sia su Windows che su Linux.

Cominciai ad interessormi anche di hordware e di aggiarnomenta delle mie ormai diventate 6 macchine con nuava RAM, schede grafiche più potenti, etc...

Per il mio 79^ compleanna, nel 2012, mi regalai i pezzi per costruirmi da sola un camputer con CPU a 8 care, una scheda grafica "da paura" e tanta, tantissimo RAM!

Ben presto rimassi quello che restava di Windows sul Campaq, ormai vecchio di 12 anni, che ora esegue una tranquilla distribuziane MX-14 e lo uso come server casolingo.

E' incredibile quanto ha imporato con Linux dal 2003 in poi! Invece di usare il PC passivamente, come un televisore o come Windows vuole che tu lo utilizzi, ara ha un ruolo ottiva e non permetterò a Microsaft o o qualcun oltra di influenzore le mie decisiani informatiche!

E' stata l'inizia del mia pensianamenta attiva!

(=)





Nel prossimo numero l'ultima creatura di Chuck Peddle: il sistema professionale Sirius-1/Victor 9000